Storage Management Component (SMC)

Configuration and Administration Guide

Release 5.1

313486102

Proprietary Information Statement

This document and its contents are proprietary to Storage Technology Corporation and may be used only under the terms of the product license or nondisclosure agreement. The information in this document, including any associated software program, may not be reproduced, disclosed or distributed in any manner without the written consent of Storage Technology Corporation.

Limitations on Warranties and Liability

This document neither extends nor creates warranties of any nature, expressed or implied. Storage Technology Corporation cannot accept any responsibility for your use of the information in this document or for your use of any associated software program. You are responsible for backing up your data. You should be careful to ensure that your use of the information complies with all applicable laws, rules, and regulations of the jurisdictions in which it is used.

Warning: No part or portion of this document may be reproduced in any manner or in any form without the written permission of Storage Technology Corporation.

Restrictive Rights

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227─7013 or subparagraphs (c) (1) and (2) of the Commercial Computer Software — Restricted Rights at 48 CFR 52.227─19, as applicable.

First Edition, December 2002
Part Number 313486102
EC 128601

This edition applies to Release 5.1 of the Nearline Control Solution (NCS) software. Information in this publication is subject to change. Comments concerning the contents of this publication should be directed to:

Storage Technology Corporation
Manager, Software Information Development
One StorageTek Drive
Louisville, Colorado 80028-5209

or

sid@stortek.com

Document Effectivity

EC Number	Date	Doc Kit Number	Type	Effectivity
128601	December, 2002	---	First Edition	This document applies to SMC, Release 5.1.

Contents

What's New With This Release?

SMC Release 5.1 includes the following changes and enhancements:

- SMP/E APPLY and ACCEPT installation steps for SMC are now included in the NCSAPPLY and NCSACCPT sample members, respectively.
- If SMC is not active on an MVS host, allocation influence does not occur, SMS esoteric substitution is not performed, and MVS messages are not intercepted.
- The SMC performs all SMS esoteric substitution for the NCS solution.
- The SMS parameter is added for the ALLOCDEF and ALLOCJOB operator commands. This parameter is used to enable or disable the DFSMS interface.
- The SMC intercepts MVS messages related to mount, dismount, and swap, and directs mount and dismount requests to the appropriate library subsystem (HSC or MVS/CSC) on the host.
    - The USERMsg operator command allows specification of additional messages to be intercepted and passed to the library subsystem's user exit 01. This command can also be used to list the message identifiers for those messages that the SMC intercepts by default.
- The TRace operator command is updated to allow tracing of individual components within jobs or the SMC subsystem.
- The LIst operator command no longer requires a Length parameter when Address is specified. When no length is provided, a default value of 100 (decimal 256) is used.
- SMC drive prioritization now allocates drives by rotation to reduce excessive wear on any one particular drive.
- The NOSEPpol parameter is no longer supported for the ALLOCDef or ALLOCJob operator commands.
- For drives outside the library, the SMC recognizes drive characteristics specified in UNITATTR statements, even if these drives are not defined in nonlibrary esoterics (SLILIBRY NNLBDRV parameter).
- For Tape Management Systems that supply a subpool, this subpool is interpreted by the SMC and used as the requested subpool name, unless overridden by user exit 01 or a TAPEREQ statement.
- Message changes, additions and deletions.

About this Guide

Intended Audience

This guide provides administration and configuration information for the Storage Management Component (SMC) software. It is intended for system programmers and operators responsible for configuring and maintaining the SMC software at their site.

How this Guide is Organized

This guide contains the following chapters:

- Chapter 1, “Introduction” describes the SMC product; its general features and functions.
- Chapter 2, “Starting the SMC” describes how to initialize the SMC software.
- Chapter 3, “SMC Allocation in a JES2 Environment” describes the SMC allocation function for a JES2 environment.
- Chapter 4, “SMC Allocation in a JES3 Environment” describes the SMC allocation function for a JES3 environment.
- Chapter 5, “SMC Message Handling” describes the SMC message intercept function.
- Chapter 6, “Operator Commands” describes the SMC operator commands.
- Chapter 7, “Recovery Procedures” describes procedures used when SMC or a library subsystem (HSC or MVS/CSC) becomes inactive.
- Chapter 8, “JES3 Configuration Report Utility” describes the SMC JES3 Configuration Report utility.
- Chapter 9, “SMC Messages” describes the SMC system messages.
- Appendix A, “JES2 User Exit Return Codes and Library Subsystem Ownership” describes JES2 user exit return codes with regard to library subsystem ownership.
- Appendix B, “JES3 User Exit Return Codes and Library Subsystem Ownership” describes JES3 user exit return codes with regard to subsystem ownership.
- Appendix C, “Intercepted Messages” lists MVS, JES3, TMS and DFSMSrmm messages that are intercepted by the SMC.

- Appendix D, “Message Change Summary” lists SMC messages that have been added, changed or deleted for this release.

An index is also included.

Conventions Used in this Guide

Product Names

HSC refers to the MVS implementation of StorageTek’s Host Software Component. The VM implementation of HSC is not supported.

MVS/CSC refers to StorageTek’s Client System Component for MVS.

Typographic

In the JCL examples in this guide, some fields appear in lower case. You must update these fields to match your installation requirements.

Symbols

The following symbols are used to highlight text in this guide:

Note: Information that may be of special interest to you. Notes are also used to point out exceptions to rules or procedures.

Warning: Information necessary to keep you from damaging your hardware or software.

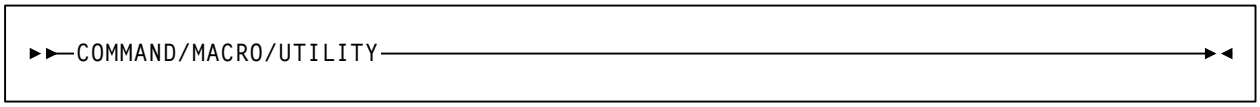
Caution: Information necessary to keep you from corrupting your data.

Syntax Flow Diagrams

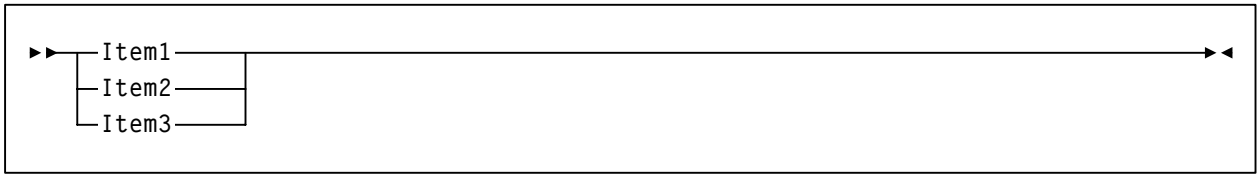
Syntax flow diagramming conventions include the following:

Flow Lines

Syntax diagrams consist of a horizontal base line, horizontal and vertical branch lines, and the text for a command, control statement, macro, or utility.



or

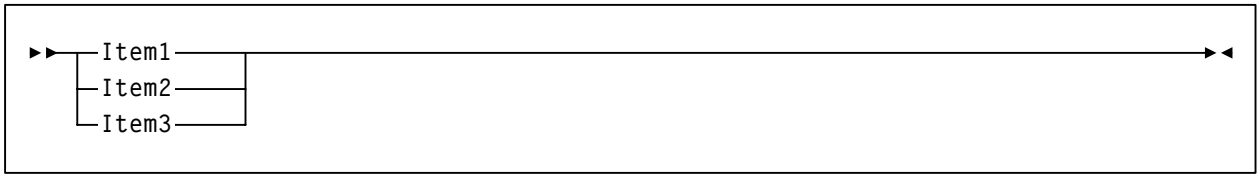


Diagrams are read left to right and top to bottom. Arrows indicate flow and direction.

- a statement begins with ►►
- a statement ends with ►◄
- diagrams continuing to the next line begin with ►
- fragments begin and end with |

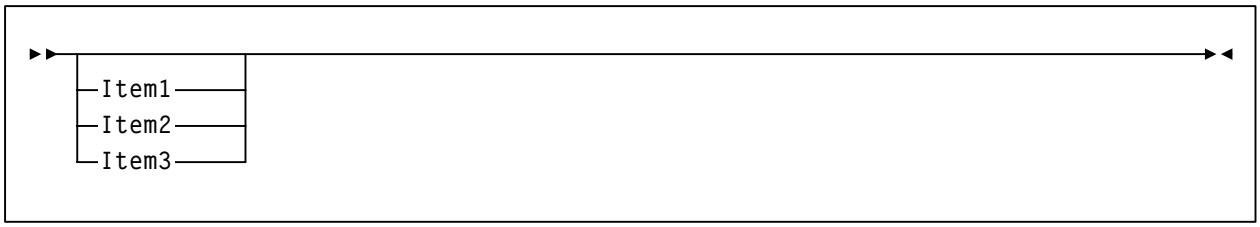
Single Required Choice

Branch lines (without repeat arrows) indicate that a single choice must be made. If one of the items from which a choice is being made is positioned on the base line of the diagram, a single choice is required.



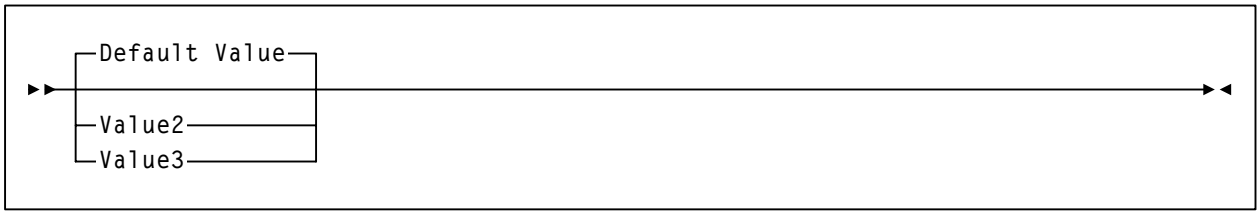
Single Optional Choice

If the first item is positioned on the line below the base line, a single choice of items in the stack is optional.

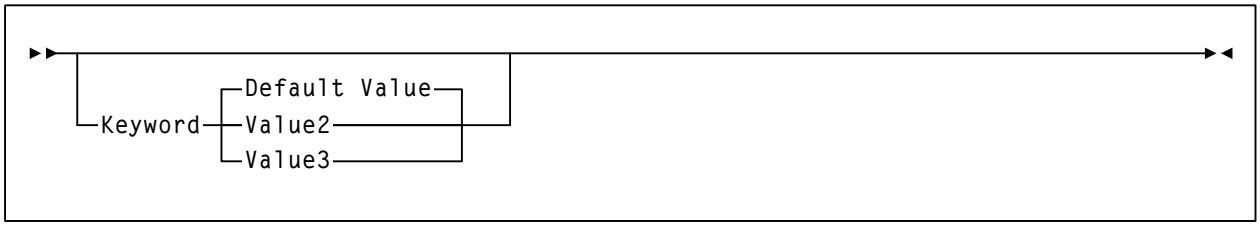


Defaults

Default values and parameters appear above the base line. In the following example, if a value is not specified with the command, the `Default Value` is used.

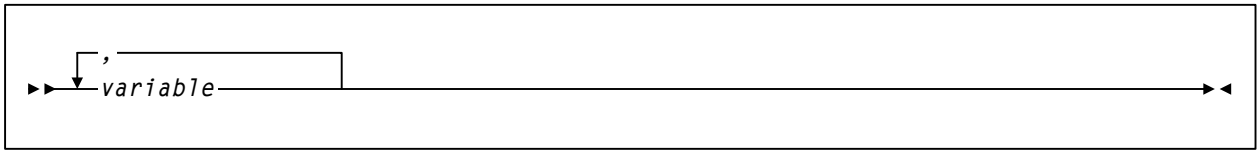


Some keyword parameters provide a choice of values in a stack. When the stack contains a default value, the keyword and the value choices are placed below the base line to indicate that they are optional, and the default value appears above the keyword line. In the following example, if the keyword is not specified with the command, the `Default Value` is used.



Repeat Symbol

A repeat symbol indicates that more than one choice can be made or that a single choice can be made more than once. The repeat symbol shown in the following example indicates that a comma is required as the repeat delimiter.



Keywords

All keywords are shown in uppercase or in mixed case. When keywords are not case sensitive, mixed case implies that the lowercase letters may be omitted to form an abbreviation.

Variables

Italic type is used to indicate a variable.

Alternatives

A bar (|) is used to separate alternative parameter values.

Delimiters

If parenthesis (), a comma (,), a semicolon (;), or any other delimiter is shown with an element of the syntax diagram, it must be entered as part of the statement or command unless otherwise stated.

Ranges

- An inclusive range is indicated by a pair of elements of the same length and data type, joined by a dash. The first element must be strictly less than the second element.
- A hexadecimal range consists of a pair of hexadecimal numbers (for example, 0A2-0AD, or 000-0FC).
- A decimal range consists of a pair of decimal numbers (i.e., 1-9, or 010-094). Leading zeros are not required. The decimal portion is referred to as an incremental range. The character positions of the incremental portion of both range elements must match, and the non-incremental characters of the first element must be identical to those of the second element.

- A numeric VOLSER range (*vol-range*) consists of a pair of VOLSER elements containing a decimal numeric portion of 1 to 6 digits (for example, ABC012-ABC025, or X123CB-X277CB). The decimal portion is referred to as an incremental range. The following additional restrictions apply:
  - The character positions of the incremental portion of both range elements must match.
  - The non-incremental characters of the first element must be identical to those of the second element.
  - You cannot increment two portions of a range element. If 111AAA is the first element, you cannot specify 112AAB for the second element.
  - If a VOLSER range contains more than one decimal portion, only the right-most portion is valid as the incremental range. For example:

    A00B00 the largest range that can be specified is A00B00 through A00B99.

    A0B0CC the largest range that can be specified is A0B0CC through A0B9CC.

    000XXX the largest range that can be specified is 000XXX through 999XXX.

- An alphabetic VOLSER range (*vol-range*) consists of a pair of VOLSER elements containing an incremental portion of 1 to 6 characters (for example, 000AAA-000ZZZ, or 9AAA55-9ZZZ55). This portion is referred to as an incremental range. The following additional restrictions apply:
  - The character positions of the incremental portion of both range elements must match.
  - The non-incremental characters of the first element must be identical to those of the second element.
  - You cannot increment two portions of a range element. If 111AAA is the first element, you cannot specify 112AAB for the second element.
  - The alphabetic portion of the VOLSER range is defined as being from character A to Z. To increment multi-character sequences, each character increments to Z. For instance, ACZ is part of the AAA-AMM range. Examples are:

A00A0-A99A0	increments VOLSERs A00A0 through A09A0, then A10A0 through A99A0.
9AA9A-9ZZ9A	increments VOLSERs 9AA9A through 9AZ9A, then 9BA9A through 9ZZ9A.
111AAA-111ZZZ	increments VOLSERs 111AAA through 111AAZ, then 111ABA through 111ZZZ
999AM8-999CM8	increments VOLSERs 999AM8 through 999AZ8, then 999BA8 through 999CM8
A3BZZ9-A3CDE9	increments VOLSERs A3BZZ9 through A3CAA9, then A3CAB9 through A3CDE9
AAAAAA-AAACCC	increments VOLSERs AAAAAA through AAAAAZ, then AAAABA through AAACCC
CCCNNN-DDDNNN	increments VOLSERs CCCNNN through CCCNNZ, then CCCNOA through DDDNNN *

* **Caution:** This is a very large range.

The number of volumes in an alphabetic VOLSER range depends on the number of elements in the incrementing portion of the VOLSER range. For an A to Z range in each character position, the number of volumes can be calculated by 26 to the power of the number of positions that are being incremented.

A-Z	26^1	26
AA-ZZ	26^2	676
AAA-ZZZ	26^3	17,576
AAAA-ZZZZ	26^4	456,976
AAAAA-ZZZZZ	26^5	11,881,376
AAAAAA-ZZZZZZ	26^6	308,915,776

Lists

A list consists of one or more elements. If more than one element is specified, the elements must be separated by a comma or a blank space, and the entire list must be enclosed in parentheses.

Blanks

Keyword parameters and values may be separated by any number of blanks.

Control Statements

The standard syntax conventions for control statements are as follows:

- The only valid control statement information area is from column 1 to column 72. Columns 73-80 are ignored.
- Parameters may be separated by one or more blanks or a comma.
- A value is associated with a parameter by an equal (=) sign or by enclosing the value in parentheses, and concatenating it immediately after the parameter.
- Case (upper or lower) is ignored in actual control statements.
- Continuations are supported by including a plus (+) sign at the end of the line to be continued.
- /* and */ can be used to enclose comments in the job stream. Comments can be continued over multiple lines, but cannot be nested.
- The maximum length for a control statement is 1024 characters.

Related Publications

The following publications contain information about specific topics relating to the use of the Storage Management Component (SMC):

StorageTek Nearline Control Solution (NCS) Publications

- *NCS Installation Guide*
- *Requesting Help from Software Support*

StorageTek Host Software Component (MVS/HSC) Publications

- *MVS/HSC Configuration Guide*
- *MVS/HSC Operator's Guide*
- *MVS/HSC System Programmer's Guide*
- *MVS/HSC Messages and Codes Guide*

StorageTek LibraryStation Publications

- *LibraryStation Configuration Guide*
- *LibraryStation Operator and System Programmer's Guide*
- *LibraryStation Messages and Codes Guide*

StorageTek Client System Component (MVS/CSC) Publications

- *MVS/CSC Configuration Guide*
- *MVS/CSC Operator's Guide*
- *MVS/CSC System Programmer's Guide*
- *MVS/CSC Messages and Codes Guide*

StorageTek Virtual Storage Manager Publications

- *VTCS Installation and Configuration Guide*
- *VTCS Administration Guide*
- *VTCS Messages and Codes Guide*
- *VTCS Reference*

IBM JES3 Publications

- *MVS/ESA JES3 Initialization and Tuning Reference*
- *OS/390 JES3 Initialization and Tuning Reference*

Technical Support

StorageTek Software Support and the StorageTek Customer Resource Center (CRC) maintain information about known SMC product updates. You can contact Software Support or access the CRC for the latest information available concerning product updates (i.e. documentation, PTFs, PUTs).

See the *Requesting Help from Software Support* guide (included in the NCS package) for information about contacting StorageTek for technical support and for requesting changes to software products, or access StorageTek's CRC homepage at:

```
http://www.support.storagetek.com
```

Note: You must obtain a login ID and password in order to access the CRC. You can request a login ID and password from the CRC homepage.

Reader's Comments

We'd like to know what you think about this guide. E-mail your comments to Software Information Development directly. Our Internet address is:

```
sid@stortek.com
```

Be sure to include the number and title of the guide along with your comments.

Chapter 1. Introduction

What is SMC?

SMC, the Storage Management Component, is the interface between IBM's OS/390 and z/OS operating systems and StorageTek real and virtual tape hardware. SMC operates on both JES2 and JES3 systems and is a **required** NCS component. Its primary functions are:

- Influencing tape allocation according to hardware requirements and customer policies to ensure that appropriate tape drives are selected.
- Intercepting operating system mount, dismount, and swap messages and translating them in order to perform the required tape hardware functions.

SMC represents a rearchitecture of the NCS solution. It consolidates HSC, MVS/CSC, JES2 and JES3 MVS interface components into consistent functions.

SMC resides on the MVS host along with the HSC and/or MVS/CSC, and communicates with these products to determine policies, volume locations, and drive ownership. When multiple subsystems are active on the host, the SMC first queries the HSC for this information, followed by all active MVS/CSCs.

Note: For the purposes of this publication, **HSC** refers to the **MVS** implementation of StorageTek's Host Software Component. The VM implementation of HSC is not supported.

New Terminology

SMC introduces two important terms:

Drive Exclusion (previously referred to as device separation) refers to the SMC function of excluding drives for an allocation request based on SMC exclusion criteria.

Drive Prioritization (previously referred to as directed allocation) refers to the SMC function of influencing selection of a particular drive based on allocation criteria, including volume location.

See Chapter 3, "SMC Allocation in a JES2 Environment" and Chapter 4, "SMC Allocation in a JES3 Environment" for more information about these terms.

Chapter 2. Starting the SMC

Overview

The SMC initializes as an independent started task on the MVS host. The SMC calls on HSC and MVS/CSC for policy, volume, and drive information. Therefore, HSC 5.1 and/or MVS/CSC 5.1 **must** be active on the same host.

Note: The SMC **must** be initialized before HSC or MVS/CSC can complete initialization. These products detect the presence of an active SMC subsystem when they attempt to initialize. If the HSC or MVS/CSC is started before the SMC, the HSC or MVS/CSC issues message SLS4510A/SCS4510A, indicating that the SMC must be started before initialization can complete.

If the SMC is started without an HSC or MVS/CSC active, message SMC0049 is issued each time the SMC attempts to manage a tape allocation, indicating that SMC allocation processing has been bypassed. Start the HSC or MVS/CSC to correct this situation.

SMC startup requires the following tasks:

- creating the SMC START procedure
- executing the SMC START procedure.

Note: Refer to the *NCS 5.1 Installation Guide* for SMC installation procedures and JES3 post-installation tasks.

Creating the SMC START Procedure

An SMC START procedure must be created in the procedure library of the host system. The MVS START command invokes this catalogued procedure, thus activating the SMC with the specified startup parameter settings.

Figure 1 provides a sample SMC START procedure.

```
//yourprocname   PROC PRM='WARM'
//stepname  EXEC PGM=SMCBINT,REGION=4M,TIME=1440,
//          PARM='&PRM'
//*
//STEPLIB   DD DISP=SHR,DSN=linklib_name
//*
//SMCPARMS  DD DSN=parmlib_name(parm_member_name),DISP=SHR
//*
//SMCCMDS   DD DSN=cmdlib_name(cmd_member_name),DISP=SHR
```

Figure 1. Sample SMC START Procedure

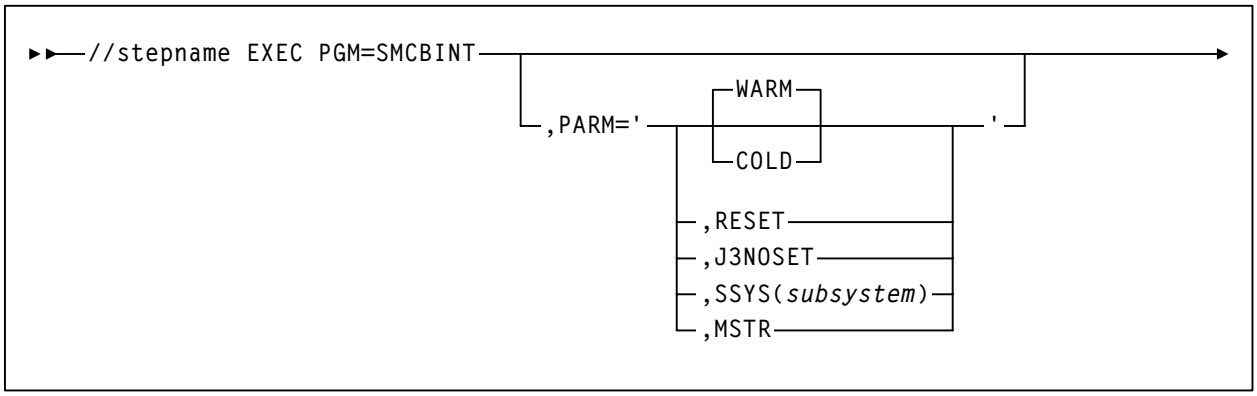
Notes:

- The first four characters of *yourprocname* specify the SMC subsystem name (unless the SSYS startup parameter is specified). StorageTek recommends a value of SMC*x*, where *x* is any valid jobname character.
- The SMCPARMS and SMCCMDS DD statements are optional. Both statements are processed during initialization. The SMCCMDS data set may be re-processed during execution. See Chapter 6, "Operator Commands" on page 109 for information about SMC commands that may be specified in either of these data sets.
- TIME=1440 must be coded to ensure that the SMC does not time out and terminate.

SMC EXEC Statement

The EXEC statement is used to define SMC startup parameter settings.

EXEC Statement Syntax



EXEC Statement Parameters

PARM=
defines the list of parameters passed to the SMC initialization routine.

Note: Execution parameters must be separated with commas. Separating parameters with blanks results in a syntax error.

WARM
specifies that the SMC main control block is not rebuilt. This is the default setting for normal operation.

COLD
specifies that all SMC control blocks are rebuilt. This parameter is mutually exclusive with WARM.

SSYS
specifies a *subsystem* ID that is different from the first four characters of the SMC START procedure. The SMC searches for this subsystem ID during initialization.

subsystem must be one to four characters in length.

RESET
specifies that the active subsystem status flag in the MVS Subsystem Communications Vector Table (SSCVT) for the SMC is reset. This parameter may correct a situation in which the SMC was terminated abnormally. It can be specified with WARM or COLD.

Warning: Using this parameter when an SMC subsystem is active and functional causes unpredictable results.

J3NOSET

indicates that a JES3 system is not using JES3 tape setup. When this parameter is specified, allocation influencing behaves as described for JES2.

MSTR

specifies that the SMC start under the MSTR subsystem instead of under JES.

When specifying this parameter, you **must** also perform **one** of the following actions:

- Start the SMC subsystem using SUB=MSTR on the MVS Start command.
- Add the SMC subsystem to the IEFSSNxx subsystem table using the keyword format.

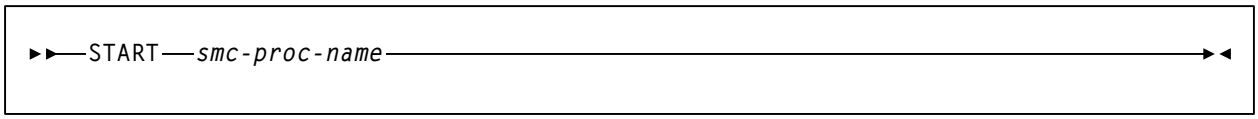
Note: This parameter is not supported for JES3 with SETUP environments.

Executing the SMC START Procedure

The MVS START command is used to initialize the SMC software. When this command is issued, the SMC subsystem initialization routine is invoked. This routine determines what parameters are in effect, performs any cleanup necessary, and begins normal processing.

Parameters associated with PARM= on the EXEC statement of the SMC Start Procedure can also be supplied via PARM= on the MVS START command. The PARM= specification on the MVS START command overrides the PARM= specification in the SMC Start Procedure. See "EXEC Statement Parameters" on page 5 for parameter descriptions.

MVS START Command Syntax



MVS START Command Parameters

START or S
initiates the MVS START command

smc-proc-name
indicates the name of the SMC START procedure member.

Chapter 3. SMC Allocation in a JES2 Environment

Overview

This chapter describes SMC allocation in a JES2 or JES3 without SETUP environment. The following topics are discussed:

- jobs selected for SMC allocation management
- DFSMS esoteric substitution
- drive exclusion
- affinity separation
- drive prioritization
- deferring mounts
- multiple library subsystem support
- SMC DFSMS processing
- interaction with other software
- differences between SMC and NCS allocation.

Jobs Selected for SMC Allocation Management

All I/O device allocations on a JES2 system are examined by the SMC to determine if it should process the allocation request.

The SMC uses the MVS subsystem interface (SSI) IEFJFRQ Subsystem Function Request exit to gain control during tape allocation events. The SMC takes control in the JES2 environment for these Subsystem Functions:

- SSI55 - DFSMS Interpreter/Dynamic Allocation Exit (IDAX)
- SSI24 - common allocation
- SSI78 - tape allocation.

SSI55 Interpreter/Dynamic Allocation Exit (IDAX)

During MVS JCL interpretation processing, IDAX provides an opportunity to examine new data set allocations and to replace the unit parameter value with a different value (esoteric).

Note: This function is optional for SMC customers. For information about implementing DFSMS esoteric substitution, refer to “SMC DFSMS Processing” on page 35.

SSI24 Common Allocation

During SSI24 common allocation processing, the SMC performs the following processes to arrive at the best set of eligible drives:

- drive exclusion
- unit affinity separation.

The results of the drive exclusion process are not reflected in MVS control blocks until tape allocation time.

The results of unit affinity separation are used to update MVS VOLUNIT entries in the SIOT.

SSI78 Tape Allocation

During SSI78 tape allocation processing, the SMC performs the following:

- updates to MVS control blocks based on drive exclusion results
- drive prioritization
- mount deferral.

The SMC sets all unacceptable drives to ineligible status and assigns a priority to each drive that remains eligible for the allocation. The higher the priority, the more likely the device will be chosen for the mount.

The mount may be deferred until the data set is opened. The customer’s library subsystem defer policy determines whether or not the mount is deferred.

The SMC implements these decisions by updating the IEFSSTA control blocks for deferral, drive exclusion, and prioritization during SSI78 processing.

Note: The SMC, unlike the HSC and MVS/CSC, does not update the Eligible Device List (EDL), unless the ALLOCDef command MIAcompat parameter is set to ON. Refer to the "ALLOCDef Command" on page 112 for more information.

Exceptions

The SMC does not influence the following types of cartridge tape allocation:

- demand allocation (i.e., request for a specific drive(s))

Note: The SMC does perform DEFER processing for demand allocation.

- allocations excluded explicitly by entering the ALLOCJob command BYPASS parameter. Refer to the description of this command in Chapter 6, "Operator Commands".
- allocations where the list of eligible devices does not contain any drives known to a library subsystem, such as drives contained in other vendor's tape libraries
- DFSMS-managed allocation. An SMS-managed data set is defined as a data set that has a storage class defined. A storage class is assigned when either
  - the STORCLAS parameter is specified on the DD statement, or
  - an installation-written ACS routine selects a storage class for a new data set.

DFSMS Esoteric Substitution

During MVS JCL interpretation processing, the Interpreter/Dynamic Allocation Exit (IDAX) provides an opportunity to examine new data set allocations and to replace the unit parameter value with a different value (esoteric).

Note: This function is optional for SMC customers. For information about implementing DFSMS esoteric substitution, refer to “SMC DFSMS Processing” on page 35.

Drive Exclusion

When the SMC determines an interest in an allocation event at common allocation time (SSI24 subsystem function), its first step is to narrow down the list of eligible drives to those that best support the allocation request. This step is called *drive exclusion*.

The SMC performs drive exclusion by building a list of drives contained in the initial EDL and then by following a step-by-step process using an ordered list of exclusion criteria to remove drives from this list.

Like the HSC and MVS/CSC device separation processes, the SMC drive exclusion process does **not** take into account the status of drives (e.g., offline, busy) when selecting drives that are eligible for the allocation request. If all drives compatible with the allocation request are unavailable, the job goes into allocation recovery.

Before any drives are removed from consideration for the allocation, the SMC first determines whether a job step can be allocated.

For example:

```
//JOB1  JOB …
//STEP1 EXEC PGM=…
//DD1   DD   UNIT=349Ø,…
//DD2   DD   UNIT=349Ø,…
          .
          .
          .
//DD25  DD   UNIT=349Ø,…
```

Assume STEP1 is requesting twenty-five 3490 drives. If the MVS EDL contains twenty-four 3490 drives in its lists of eligible devices, MVS cannot allocate the required drives for the job. In this instance, the SMC issues message SMC0042, which indicates the job cannot allocate before SMC modification, and allows the job to fail allocation.

Once the SMC determines a job step can be allocated, the SMC begins excluding drives based on an ordered list of criteria. Table 1 on page 16 and Table 2 on page 21 list these criteria and show the order the exclusion process follows. The lower the criteria level, the higher its importance. Thus, level 2 is considered more important than level 6.

Minimum Exclusion Level

By default, level 2 is the *minimum exclusion level*, that is, the desired minimum level of allocation exclusion, for both specific and scratch volume processing. However, if necessary, the ALLOCDef and ALLOCJob commands can reset the minimum exclusion level (refer to Chapter 6, "Operator Commands" for descriptions of these commands).

Note: If your VTCS configuration allows recalls of all migrated virtual volumes to any VTSS, StorageTek recommends you set the minimum exclusion level to 1 using the ALLOCDef command. See "Exclusion Level 2" on page 17 for more information.

In the first step of drive exclusion, the SMC determines if the job can be allocated at the minimum drive exclusion level. If not enough drives remain eligible at the minimum exclusion level, the SMC issues message SMC0043, which indicates the job is not allocatable at the minimum exclusion level. The SMC then excludes all devices, causing the job to fail allocation.

For example:

```
//JOB2  JOB …
//STEP1 EXEC
//DD1   DD   UNIT=3490,VOL=SER=984001,DISP=OLD
//DD2   DD   UNIT=3490,VOL=SER=984002,DISP=OLD
//DD3   DD   UNIT=3490,VOL=SER=984003,DISP=OLD
```

In this system environment, assume that twenty 3490-type drives have been configured. Only two of those twenty are actually 9840 drives. If all three volumes require a separate 9840 tape drive, then not enough 9840 drives exist for the job to allocate. Without SMC influence, MVS allows the job to proceed through allocation. However, with SMC influence, message SMC0043 is issued and the job fails.

Note: No exclusion criteria (including minimum level) are applied when the user specifies demand allocation (i.e., requests a specific drive or drives).

Subsequent Levels

Once the minimum level of exclusion has succeeded, the remaining exclusion levels are performed. Any level above the minimum level that fails is skipped and exclusion continues to the next level.

Note: A level fails if it results in too few drives for the DD statement to allocate. The only required exclusion levels are those at the minimum level or below.

After the last drive exclusion level is completed, the SMC examines the final drive lists. If the job step cannot be allocated because not enough drives remain eligible, drive exclusion regresses to the previous exclusion level for the failing DD statement. This process repeats until the job step can allocate.

Intentionally Failing a Job

Depending upon the characteristics of your installation, you may want specific jobs to fail at allocation time rather than execute them with non-optimal devices. As an example, if you have two ACSs, one of which is located at a remote site, you may prefer to fail the job, rather than having to transfer the correct volume from one ACS to another.

For example:

```
//DD1  DD  UNIT=CART,VOL=SER=INACSØ,DISP=OLD
//DD2  DD  UNIT=AFF=DD1,VOL=SER=INACS1,DISP=OLD
```

Assume the following:

- Both volume INACS0 and INACS1 share common media characteristics.
- Volume INACS0 resides in ACS0, and volume INACS1 resides in ACS1.
- Only ACS0 contains a compatible drive.

If this affinity chain is not broken, then volume INACS1 must be ejected from ACS1 and entered into ACS0 for this job to execute, which may not be possible given the geographical locations of the ACSs.

In this case, failure at allocation time may be preferable to failure at run time. By specifying the correct MINLVL value (i.e., by issuing the ALLOCJob command with a MINLVL of 7 for this job), the job is failed at allocation time.

Caution: Setting the minimum exclusion level may cause jobs to fail unexpectedly. Whenever the SMC is unable to apply an exclusion criterion at or below the minimum level, the job fails allocation.

For example, if a TAPEREQ specifies a media of STANDARD for a scratch volume and the eligible device list contains only 9840 drives, when the SMC attempts to apply the “policy media” exclusion level, all drives are excluded. If the minimum level is set to 3 or higher, the job fails allocation. Therefore, StorageTek recommends that the default exclusion level (as specified by the ALLOCDef command) should normally not be set to a level greater than 2.

Specific Volume Request Drive Exclusion Criteria

For a specific volume allocation, the SMC excludes drives in order from lowest exclusion level to highest based on the criteria in Table 1. The lower the level number, the more important the exclusion criteria.

Table 1. JES2 Drive Exclusion Levels

Level	Specific Volume Criteria	Keyword*
1	Exclude drives incompatible with the volume media. Primary source: external volume label Secondary source: VOLATTR MEDIA parameter	MEDRECTECH
2**	For virtual volumes only, exclude drives not located in an accessible VTSS. Note: No further exclusion is performed for virtual volumes.	VTSSLOCATION
3	Exclude drives based on the required recording technique. Source: VOLATTR RECTECH parameter.	VOLATTRRECTECH
4	Exclude drives based on user location policies. Primary source: TAPEREQ ESOTERIC parameter. Secondary source: HSC or MVS/CSC User Exit 08 or 10 return codes.	USERPOLICY
5	Exclude drives based on the HSC ALLOC command SPECVOL parameter. Note: This exclusion level does not apply to MVS/CSC.	SPECVOL
6	Exclude drives based on volume location type, i.e., library or nonlibrary.	LOCTYPE
7	Exclude drives based on the ACS location of the volume, if the volume resides in the library.	ACSLOCATION
8	Exclude drives based on the requested recording technique. Primary source: DFSMS data class recording technique. Secondary source: TAPEREQ RECTECH parameter.	POLRECTECH

* Keywords associated with each exclusion level are specified in the exclusion criteria displayed in message SMC0046.

** Level 2 is the default minimum level.

Exclusion Level 1

The SMC excludes drives that are not compatible with the volume media. The volume media can be obtained from the volume label or from an HSC VOLATTR statement MEDIA parameter.

Exclusion Level 2

If the specific volume requested is a virtual volume, the VTSS location for a resident volume must match the volume location. The VTSS location for a migrated virtual volume must match the location to which the volume can be recalled. The remaining exclusion levels do not apply to virtual volumes and are bypassed.

Note: If all VTSSs in your virtual configuration allow migration to common ACSs, you may want to set the minimum level exclusion level to 1 to allow jobs to proceed by migrating and recalling resident VTVs.

See the VTCS documentation for more information about virtual volumes.

Exclusion Level 3

If an HSC VOLATTR statement specified the recording technique (RECTECH parameter) for this specific volume, the SMC excludes drives that do not provide that recording technique.

For example:

```
//DD1 DD UNIT=TLACS0,VOL=SER=VOL000,DISP=OLD
```

A VOLATTR statement has been defined to the HSC.

```
VOLATTR SERIAL(VOL000) RECTECH(36B) MEDIA(ECART)
```

VOL000 requires a 9490 drive, and if TLACS0 contains both 9490 and 9490EE drives, the 9490EE drives are excluded from consideration for this allocation.

Note: In MVS/CSC, the VOLATTR RECTECH is implied by the composite recording technique of the drives in the list returned by the library server.

Exclusion Level 4

If a TAPEREQ statement specifies an esoteric unit name for this allocation request, the SMC verifies the esoteric unit name is defined to MVS. If the esoteric passes this test, any drives in the original EDL that are not also defined to the esoteric unit name are excluded from consideration for this allocation.

For example:

```
//DD2   DD UNIT=CART,VOL=SER=TR0001,DSN=SYS4.TR1.DATA,DISP=OLD
```

The following TAPEREQ statement is defined to HSC or MVS/CSC.

```
TAPEREQ DSN(SYS4.TR1.**) ESOT(TLACS1) RECT(LONGI)
```

The drives that are not defined to the esoteric TLACS1 are excluded from consideration for this allocation.

If no TAPEREQ statement specifies an esoteric for this allocation request, HSC and MVS/CSC User Exit 08 or 10 return codes and values can define the criteria for drive exclusion at this level. The user exit return codes can direct the SMC to select nonlibrary drives, a specific ACS, or the user exit can return an esoteric name that contains customer-selected drives. Any drives in the current list of eligible devices that did not match the user exit return values are excluded here.

For example:

```
//DD1    DD UNIT=CART,VOL=SER=VOL001,DISP=OLD
```

HSC User Exit 08 returns UX08ASUB and ACS0, which requests library drives and ACS0. The drives that are not in ACS0 are excluded from consideration for this allocation.

Exclusion Level 5

The SPECVOL parameter of the HSC ALLOC command instructs the SMC which library ACS locations can be used for this request if one of two circumstances exists:

- the specific volume is not in the library and no drives are defined in the nonlibrary esoteric to the HSC, or
- the specific volume is not in the library, and user exit 08 requested a library location.

Drives not in SPECVOL ACS locations are excluded from consideration.

Note: This exclusion level does not apply to MVS/CSC.

For example:

```
//DD1   DD UNIT=CART,VOL=SER=NL0001,DISP=OLD
```

Volume NL0001 is outside the library, and the HSC command ALLOC SPECVOL(00) has been issued.

If only library drives are defined to the HSC, only the drives defined to ACS00 remain under consideration for this allocation.

Another example:

```
//DD2   DD UNIT=CART,VOL=SER=NL0002,DISP=OLD
```

Assume volume NL0002 is not in the library, and the HSC command ALLOC SPECVOL ACS(01) has been issued to allow ACS01 to be used for specific nonlibrary volumes.

If nonlibrary drives have been defined on the HSC LIBGEN SLILBRY macro but HSC user exit 08 returned UX08SVOL, all nonlibrary and drives in ACS00 are excluded from consideration for this allocation.

Exclusion Level 6

The generic location of the volume (library or nonlibrary) reduces the remaining list of eligible drives.

Note: For exclusion level 6, nonlibrary drives are outside the library and contain **either** known (from the HSC NNLBDRV or MVS/CSC NONLIB esoteric) or unknown device characteristics.

If the volume resides in a library, all drives outside the library are excluded, and if the volume resides outside the library, all library drives are excluded.

For example:

```
//DD1   DD UNIT=CART,VOL=SER=LIB001,DISP=OLD
```

Assume volume LIB001 resides in a library.

All drives outside the library are excluded from consideration for this allocation.

Exclusion Level 7

For volumes residing in the library, the ACS location of the volume reduces the remaining list of eligible drives. Any drives that remain that do not reside in the same ACS as the volume are excluded.

For example:

```
//DD1   DD UNIT=CART,VOL=SER=A00VL1,DISP=OLD
```

Assume volume A00VL1 resides in ACS00, and the drives reside in ACS00 and ACS01.

The drives residing in ACS01 are excluded from consideration for this allocation.

Exclusion Level 8

Exclusion levels 1 and 3 restrict the list of eligible drives to those compatible with the volume's actual media and its recording technique, if specified on the HSC or MVS/CSC server VOLATTR statement. Exclusion level 8 may further restrict the drives for the request based on the TAPEREQ or DFSMS data class recording technique.

The HSC or MVS/CSC TAPEREQ definitions can explicitly denote a recording technique for the allocation request. DFSMS data class definitions can also specify an 18- or 36-track recording technique for the request and override any TAPEREQ recording technique specification.

For example:

```
//DD1   DD UNIT=CART,VOL=SER=TV9840,DISP=NEW,
//         DSN=SYS4.TR4.DATA
```

Assume the requested volume is a 9840 volume in the library with no recording technique specified on a VOLATTR. Assume an HSC TAPEREQ has been defined:

```
TAPEREQ DSN(SYS4.TR4.**) RECT(STK1RB)
```

Previous exclusion levels restricted the list of devices to all 9840 models. In this example the TAPEREQ recording technique of STK1RB leaves only 9840B devices in consideration for this allocation.

Scratch Volume Request Drive Exclusion Criteria

For a scratch volume allocation, the SMC excludes drives in order from lowest exclusion level to highest based on the criteria in Table 2. The lower the level number, the more important the exclusion criteria.

Table 2. JES2 Drive Exclusion Levels

Level	Scratch Volume Criteria	Keyword*
1	For nonlabeled (NL) scratch volume requests, exclude all virtual drives.	VIRTUALLABEL
2**	No level 2 for scratch volume exclusion.	
3	Exclude drives based on the requested media. Primary source: DFSMS data class media specification. Secondary source: TAPEREQ MEDIA parameter. Tertiary source: HSC or MVS/CSC User Exit 02 request for virtual media.	POLMEDIA
4	Exclude drives based on user location policies. Primary source: TAPEREQ ESOTERIC parameter. Secondary source: HSC and MVS/CSC User Exit 02 and 10 return codes.	USERPOLICY
5	Exclude drives based on the media of available scratch volumes in subpool. Primary source: TAPEREQ SUBPOOL parameter. Secondary source: HSC or MVS/CSC User Exit 02 subpool values. Tertiary source: scratch subpool 0.	SUBPOOL
6	Exclude drives based on available scratch location type, i.e., library, nonlibrary, or virtual.	LOCTYPE
7	Exclude drives based on the HSC ALLOC or MVS/CSC ALTER command ZEROSCR parameter.	ZEROSCRATCH
8	Exclude drives based on the requested recording technique. Primary source: DFSMS data class recording technique. Secondary source: TAPEREQ RECTECH parameter.	POLRECTECH

* Keywords associated with each exclusion level are specified in the exclusion criteria displayed in message SMC0046.

** The MINLVL parameter value default, level 2 on the ALLOCDef or ALLOCJob command, applies to both scratch and specific volume requests. Even though level 2 has no meaning for scratch, level 2 is considered the default minimum level.

Exclusion Level 1

The SMC excludes virtual drives if the volume label type requested is NL (nonlabeled). This is the only required criterion. Refer to the VTCS documentation for more information about virtual volumes.

Exclusion Level 2

No level 2 scratch volume exclusion is performed.

Exclusion Level 3

Under HSC and MVS/CSC, a DFSMS data class definition can explicitly request the media desired, or the media can be derived from the DFSMS data class recording technique. Drives that do not provide the requested media support are excluded at this level.

If no DFSMS data class definition applies for this allocation, the HSC or MVS/CSC TAPEREQ definitions can explicitly specify the media desired for the allocation request, or media can be derived from the TAPEREQ RECTECH. Drives that do not provide the requested media support are excluded at this level.

For example:

```
//DD1  DD UNIT=CART,DISP=(NEW,KEEP),DSN=SYS4.VIRT.DATA
```

No HSC TAPEREQ has been defined and no DFSMS data class definition applies. Assume the HSC User Exit 02 return code requested virtual media (UX02VIRT). The HSC User Exit 02 media selected for this request is virtual. Only virtual drives remain in consideration for this allocation. The next example shows how TAPEREQ can override User Exit 02 return codes.

For example:

```
//DD1   DD UNIT=CART,DISP=(NEW,KEEP),DSN=SYS4.TR4.DATA
```

An HSC TAPEREQ has been defined:

```
TAPEREQ DSN(SYS4.TR4.**) MEDIA(DD3A) RECT(DD3)
```

Assume HSC User Exit 02 return code requested virtual media (UX02VIRT).

If no DFSMS data class definition or TAPEREQ definition applies to this allocation request, HSC or MVS/CSC User Exit 02 return code, UX02VIRT, can explicitly request virtual media. Non-virtual drives are excluded at this level.

However, the TAPEREQ media definition overrides the user exit return code, and the media selected for this request is helical. Only helical drives remain in consideration for this allocation. Messages SMC0045 and SMC0046 are issued indicating that the user exit request for virtual volumes was not honored.

Exclusion Level 4

If a TAPEREQ statement specifies an esoteric for this allocation request, the SMC determines the common drives between the TAPEREQ esoteric and the list of remaining eligible drives. If common drives exist, any drives remaining in the current list of eligible drives that are not also defined to the esoteric are excluded from consideration for this allocation.

If no TAPEREQ statement specifies an esoteric for this allocation request, HSC or MVS/CSC User Exit 02 and User Exit 10 return codes and values can define this criterion for drive exclusion. The user exit return codes can direct the SMC to select library drives, nonlibrary drives, or a specific ACS, or the user exit can return an esoteric unit name that contains customer-selected drives. Any drives in the current list of eligible drives that did not match the user exit return values are excluded here.

For example:

```
//DD1    DD UNIT=CART,DSN=&&TEMP1,DISP=(NEW,PASS)
```

Assume HSC User Exit 02 returns UX02LIB, which requests library drives.

The drives that are not in the library are excluded from consideration for this allocation.

A TAPEREQ example is:

```
//DD2   DD UNIT=CART,DSN=SYS4.TR1.DATA,DISP=(NEW,KEEP)
```

The following TAPEREQ statement is defined to the HSC or MVS/CSC.

```
TAPEREQ DSN(SYS4.TR1.**) ESOT(TLIB9840) RECT(LONGI)
```

The drives that are not defined to the esoteric TLIB9840 are excluded from consideration for this allocation. Any User Exit 02 return codes would be ignored.

Exclusion Level 5

The combined media and specified recording technique of the volumes in a scratch subpool define this criterion for drive exclusion. An HSC or MVS/CSC TAPEREQ SUBPOOL parameter can specify a scratch subpool name for the request.

If no TAPEREQ SUBPOOL parameter is specified for this allocation, HSC or MVS/CSC User Exit 02 can return a scratch subpool number or subpool name when it also sets the return code to use default processing or library drives. HSC or MVS/CSC User Exit 10 can also apply to affinity chains here. Refer to “Affinity Separation” on page 27 for a discussion about the interaction between User Exit 02 and User Exit 10.

Note: When no specific subpool applies to the allocation, the default subpool, subpool 0, is used.

Drives that do not provide a recording technique compatible with a volume(s) in the subpool are excluded from consideration for this allocation.

For example:

```
//DD1    DD UNIT=CART,DSN=NEW.TRK36.DATA,DISP=(NEW,KEEP)
```

Assume HSC has scratches of media types 36TRACK and DD3 in a subpool named SUBPOOL3, and a TAPEREQ entry exists:

```
TAPEREQ DSN(NEW.*) SUBPOOL(SUBPOOL3)
```

Drives providing 36-track recording and helical recording remain in consideration for this allocation.

A User Exit 02 example for this criteria:

```
//DD1   DD UNIT=CART,DSN=SYS4.MYGROUP.DATA,DISP=(NEW,KEEP)
```

Assume the following:

- HSC User Exit 02 returns UX02LIB
- HSC User Exit 02 returns MYSUBPL in UX02SNAM
- MYSUBPL has been defined to the HSC using a SCRPOOL statement in the SLSSYSxx PARMLIB member or SCRPDEF file
- HSC has scratch volumes defined in MYSUBPL scratch subpool.

The drives that are not in the library are excluded from consideration for this allocation during level 4 drive exclusion.

Exclusion Level 6

The generic location of available scratch volumes provides the next level of drive exclusion. If the library contains no scratch volumes, all library drives and virtual drives are excluded from selection, leaving only nonlibrary drives eligible. If the request is for a virtual scratch volume, eligible virtual drives are selected based on the management class from DFSMS or TAPEREQ; VTSSs that cannot support the requested management class migration are excluded. Lastly, if the library contains scratch volumes and the request is not for a virtual volume, then all virtual drives and all nonlibrary drives are excluded, leaving only library drives eligible.

For example:

```
//DD1   DD UNIT=CART,DSN=MY.NEW.DATASET,DISP=(NEW,KEEP)
```

Assume the HSC is active, there are no MVS/CSCs, the HSC has no nonlibrary drives defined, and the library contains no scratch volumes.

All HSC library drives are excluded from consideration for this allocation. When all drives are excluded during an exclusion level, and the level is not at or below minimum level, then the following occurs:

- messages SMC0045 and SMC0046 are issued
- all drives excluded at that level are restored to the list of eligible drives
- processing continues at the next level of exclusion.

In this case, all the HSC drives that were excluded by this criterion are now restored to the list. Processing continues at level 7.

Exclusion Level 7

The HSC ALLOC command and MVS/CSC ALTER command or startup parameter, ZEROSCR, determines whether or not to include drives in ACSs that do not contain any scratch cartridges of the required type. ZEROSCR(OFF), the default, indicates to include drives in all ACSs whether they contain scratch cartridges or not. ZEROSCR(ON) excludes drives in ACSs that do not contain any scratch volumes.

For example:

```
//DD1   DD UNIT=CART,DSN=MY.NEW.DATASET,DISP=(NEW,KEEP)
```

Assume that ZEROSCR(OFF) has been specified, and two ACSs have scratch counts as follows: ACS0 has 400 and ACS1 has 0.

Drives in ACS0 and ACS1 remain eligible for the allocation.

Another example is:

```
//DD2   DD UNIT=CART,DSN=MY.OTHER.DATASET,DISP=(NEW,KEEP)
```

Assume ZEROSCR(ON) has been specified, and three ACSs have scratch counts as follows: ACS0 has 400, ACS1 has 500, ACS2 has 0.

Drives in ACS0 and ACS1 remain eligible. ACS2 drives are excluded from consideration because ACS2 does not contain any scratch volumes.

Exclusion Level 8

Exclusion level 3 restricts the list of eligible drives to those compatible with the TAPEREQ or DFSMS requested media. Exclusion level 5 restricts the list of eligible drives to those compatible with the scratch media in the requested subpool. Exclusion level 8 may further restrict the devices for the request based on the TAPEREQ or DFSMS data class recording technique.

The HSC or MVS/CSC TAPEREQ definitions can explicitly specify a recording technique for the allocation request. DFSMS data class definitions can also denote an 18- or 36-track recording technique for the request and will override any TAPEREQ recording technique specification.

This criterion is the last to be applied to the drives that are eligible for the allocation. Any drive that does not provide the requested recording technique is excluded from consideration for the allocation.

For example:

```
//DD1   DD UNIT=CART,DISP=(NEW,KEEP),DSN=SYS4.TR4.DATA
```

Assume an HSC TAPEREQ has been defined:

```
TAPEREQ DSN(SYS4.TR4.**) RECT(STK1RB)
```

Previous exclusion levels restricted the list of devices to all 9840 models. In this example, the TAPEREQ recording technique of STK1RB leaves only 9840B devices in consideration for this allocation.

Affinity Separation

Explicit unit affinity is an MVS facility that allows volumes associated with two separate JCL DD statements, or allocation requests, to be mounted serially on the same drive. A request for all generations of a GDG group (GDG ALL chain) can be considered as a GDGALL affinity.

Unlike HSC and MVS/CSC, SMC makes no distinction between these two types of affinity. When processing an affinity chain begins, the drive exclusion process examines each allocation in the chain separately up to and including the minimum exclusion level. The chain is always separated when the minimum exclusion level processing results in lists of eligible drives, for two or more members of the chain, that do not contain common drives.

For example:

```
//DD1   DD UNIT=CART,DSN=MY.HELICAL.DATASET,DISP=OLD
//DD2   DD UNIT=AFF=DD1,DSN=MY.LONGI.DATASET,DISP=OLD
```

DD1 specifies a data set on helical media and DD2 specifies a data set on longitudinal media. Drive exclusion level 1 for specific volumes creates a list of eligible drives for each DD according to volume media required. The two lists do not contain a common drive. As a result, DD1 and DD2 no longer represent one drive allocation but two separate allocation requests. At this point, the SMC breaks the affinity chain between them.

Affinity Head-Of-Chain

For SMC affinity chain processing, the "head" of the affinity chain containing only scratch or only specific volumes is the first DD statement in the chain. If an affinity chain contains both scratch and specific volumes, the first specific volume is the "head."

User Policy Influence on Affinity Separation

After the minimum level of drive exclusion and affinity separation completes, user policy influences the remaining affinity separation decisions.

Further affinity chain processing decisions are based on the first value found between the following:

- HSC or MVS/CSC User Exit 10 return code separation decision
- ALLOCDef or ALLOCJob command SEPLvl parameter value.

User Exit 10

HSC or MVS/CSC User Exit 10 sends an entire affinity chain to the user for examination. The user can decide whether the affinity chain remains together or separates during exclusion processing after the minimum level.

If User Exit 10 returns UX10SEP, indicating the chain members can be separated, the SMC proceeds as if ALLOCDEF SEPLVL=MAX were specified.

If User Exit 10 returns UX10LDRV, UX10NDRV, UX10ESUB, or UX10ASUB, the affinity chain remains together for the remainder of drive exclusion processing. The location requested (e.g., library drives if UX10LDRV is returned) replaces any location value provided by HSC or MVS/CSC User Exits 02 or 08. This location information is used by drive exclusion level 4. Refer to Table 1 on page 16 and Table 2 on page 21 for drive exclusion level information.

For example:

```
//DD1   DD UNIT=CART,DSN=SYS4.DATASET1,VOL=SER=NOLIB1,DISP=OLD
//DD2   DD UNIT=AFF=DD1,DSN=SYS4.DATASET2,VOL=SER=INLIB2,DISP=OLD
```

Assume the following:

- DD1 requests a nonlibrary volume and DD2 requests a library volume.
- User Exit 08 returns a nonlibrary esoteric for DD1 (UX08ESUB).
- User Exit 10 returns a no separate decision and a library location (UX10LDRV).

The affinity chain stays together. Drive exclusion ignores the esoteric returned by User Exit 08 for DD1 and excludes all but library drives as requested by User Exit 10.

Any one of the four return values UX10LDRV, UX10NDRV, UX10ESUB, and UX10ASUB from User Exit 10 causes the SMC to operate as if ALLOCDEF SEPLvl=MIN were specified.

Note: Any HSC or MVS/CSC User Exit 10 separation decision overrides the SEPLvl parameter of the ALLOCDef or ALLOCJob commands.

SEPLvl Parameter Value

If the decision to separate or keep the affinity chain together still has not been made, further affinity chain processing depends on the SEPLvl parameter value of the ALLOCDef and ALLOCJob commands (refer to Chapter 6, “Operator Commands,” for descriptions of these commands and setting an affinity separation level).

The SEPLvl parameter operates in three distinct ways depending on the value specified. The SEPLvl MAX parameter allows drive exclusion to be performed to the maximum level for each allocation element in the affinity chain. At the end of the drive exclusion process, if two or more elements of the chain no longer have common drives, the affinity chain is separated.

For example:

```
//DD1   DD UNIT=349Ø,VOL=SER=INACSØ,DSN=MY.DSN1,DISP=OLD
//DD2   DD UNIT=AFF=DD1,VOL=SER=INACS1,DSN=MY.DSN2,DISP=OLD
```

Assume the following:

- Both volume INACS0 and INACS1 share common media characteristics.
- Volume INACS0 resides in ACS0, and volume INACS1 resides in ACS1.
- ALLOCDEF SEPLVL(MAX) has been specified.

During specific volume exclusion level 7, the list of eligible drives in DD1 are all located in ACS0 and the list of eligible drives in DD2 are all located in ACS1. No drives are common to both lists. This affinity chain separates.

Note: MAX is the default value for SEPLvl.

The SEPLvl parameter can be specified as MIN. The value MIN requests that any affinity chains that remain together after the minimum level of drive exclusion should not be separated during the remaining levels of drive exclusion processing.

Using the same example:

```
//DD1   DD UNIT=349Ø,VOL=SER=INACSØ,DSN=MY.DSN1,DISP=OLD
//DD2   DD UNIT=AFF=DD1,VOL=SER=INACS1,DSN=MY.DSN2,DISP=OLD
```

Once again, assume:

- Both volume INACS0 and INACS1 share common media characteristics.
- Volume INACS0 resides in ACS0, and volume INACS1 resides in ACS1.
- ALLOCDEF SEPLVL(MIN) has been specified.

During specific volume exclusion level 7, the list of eligible drives for DD1 is all located in ACS0. DD1 is considered the “head” of the chain and exclusion is performed to ACS0. Only drives in ACS0 remain eligible for the allocation, and the affinity chain remains together.

The SEPLvl parameter can also be set to a specific drive exclusion level. The level number indicates the highest drive exclusion level at which separation can occur. Using the previous example, if the user specifies

```
ALLOCDEF SEPLVL=6
```

The chain remains together through drive exclusion level 6 because each level produces a drive list containing common drives for both DD1 and DD2.

Another example:

```
//DD1  DD UNIT=349Ø,VOL=SER=INACSØ,DSN=VIRTUAL.DATASET1,DISP=OLD
//DD2  DD UNIT=AFF=DD1,VOL=SER=INACS1,DSN=VIRTUAL.DATASET2,DISP=OLD
//DD3  DD UNIT=AFF=DD1,VOL=SER=NL0ØØ1,DSN=REAL.DATASET,DISP=OLD
```

Assume the following:

- ALLOCDEF SEPLVL=5 has been specified.
- Volume INACS0 is in ACS0 and VOLATTR MEDIA(ECART) RECTECH(36B) has been defined.
- Volume INACS1 is in ACS1 and VOLATTR MEDIA(ECART) RECTECH(36B) has been defined.
- Volume NL0001 is a nonlibrary volume and VOLATTR MEDIA(DD3A) has been defined.

At drive exclusion level 1, two chains result because of incompatible media (ECART and DD3A). Statements DD1 and DD2 are still chained together and continue through exclusion levels 2 through 5 because the exclusion criteria do not create disparate sets of eligible drives for the two DD statements.

No further separation of the DD1/DD2 chain can occur because the SEPLVL parameter has been set to level 5.

After the last drive exclusion level is completed for the job, the SMC examines the final drive lists. If the job cannot be allocated because not enough drives remain eligible, drive exclusion regresses to the previous exclusion level of the failing DD statement and a separated chain may be rejoined. This process repeats until the job can allocate.

Note: The SEPLvl parameter replaces the function of the HSC ALLOC and MVS/CSC ALTER command UNITAFF and GDGALL parameters. These parameters are no longer supported.

If your installation previously specified UNITAFF(NOSEP) and GDGALL(NOSEP), to preserve that behavior, specify a SEPLvl value of 3 on the ALLOCDef command. If your installation previously specified UNITAFF(SEP) and GDGALL(SEP), no SEPLVL parameter change is required, since the default SMC behavior separates affinity chains at the maximum separation level.

Refer to the ALLOCDef and ALLOCJob commands in Chapter 6, “Operator Commands,” for more information about setting an installation default minimum exclusion level and unit affinity separation level.

Drive Prioritization

After drive exclusion has occurred during the SSI24 common allocation event, the SMC then proceeds to assign a priority value to each remaining eligible drive during the corresponding SSI78 tape allocation event.

Note: Between a common allocation event (SSI24) for a tape and its tape allocation event (SSI78), MVS may make drives ineligible. When this occurs, the SMC retests the job step to ensure that it is allocatable with the remaining drives. If it is not, drives that were excluded by the SMC SSI24 process are reincluded until the job step is allocatable.Only the remaining drives are considered when the SMC assigns a priority value at SSI78 time.

One exception to the reinclusion process occurs when the ALLOCDef MIAcompat parameter is set to ON. In this case, exclusion is already done, so drive reinclusion cannot take place, and the job is allowed to fail.

The SMC drive priority is assigned based on the following criteria:

- For specific volumes, drives in LSMs closest to the volume are preferred.
- For scratch volumes, drives in LSMs with the largest number of scratch volumes are preferred.
- The TAPEREQ DEVTPREF parameter causes drives with specific recording techniques to be preferred.
- The TAPEREQ SCOPE parameter determines the relative priority of LSM location and drive type preferencing.

The following parameters, control statements, and user exits also influence the drive priority:

- HSC ALLOC command parameters LOWSCR, LSMPREF, and SCRTECH
- HSC SLSUX02 user exit.

After a final list of drives has been selected for allocation, the preference order of the eligible drives, after considering LSM and drive type preferencing, is selected based on a “last use” algorithm.

To reduce excessive wear on allocated drives, the SMC allocates drives by rotation based on the “last mount time” for each drive. This value is examined for every drive in the final drive list. The drive that had the most recent mount is located, and the drive immediately following it in the list is selected as the most preferred for the current allocation.

Note: This algorithm does not apply to virtual drives.

Deferring Mounts

The SMC can also set the deferred mount status if the installation chooses to defer the mount of library volumes.

The HSC ALLOC and MVS/CSC ALTER command DEFER parameter can set defaults for an installation. The following list describes the DEFER parameter settings and how the value determines if mounts are deferred. The HSC value is listed first followed by the corresponding MVS/CSC value.

Defer(OFF/NO)
The volume mount is deferred only if the user's JCL requests it. This is the default.

Defer(ON/YES)
The volume mount is deferred until the data set is opened. HSC SLSUX09 or MVS/CSC SCSUX09 can be used to override this value on an individual mount basis.

For optimal performance, StorageTek recommends setting Defer(ON/YES).

Note: Virtual mounts are always deferred.

Multiple Library Subsystem Support

When the SMC runs on an MVS host with multiple active library subsystems (i.e., HSC and/or MVS/CSCs) at the required NCS 5.1 or later release level, the SMC determines a library subsystem "owner" for each request.

The SMC builds and maintains a list of eligible library subsystems. The HSC subsystem, if active, always appears at the beginning of the list followed by all active MVS/CSC subsystems. Each subsystem is queried in list order to select the owner of the cartridge drive allocation. The following criteria, in the order listed, determine the "ownership" of the request.

1. Original Eligible Device List (EDL)
2. TAPEREQ esoteric
3. Subsystem User Exits 02 and 08
4. Specific volume location
5. Scratch volume availability.

If after evaluating all five criteria, no owner of the allocation request has been determined, the first subsystem in the available library subsystem list becomes the owner by default. The five criteria are explained below.

Original Eligible Device List (EDL)

The SMC examines the list of devices in the original EDL. If no drives in the EDL are owned by a subsystem, that subsystem is excluded from consideration.

For example:

```
//DD1 DD DSN=THIS.DATASET,VOL=SER=CSC001,UNIT=HSCDRVS
```

Assume the following:

- Volume CSC001 resides in a library controlled by a MVS/CSC subsystem.
- All drives defined in esoteric HSCDRVS reside in a library controlled by the HSC subsystem.
- Both MVS/CSC and HSC are active.

The HSC owns the allocation request because it owns the drives.

TAPEREQ Esoteric

When the list of devices does not all belong to a single subsystem, a TAPEREQ esoteric can direct the allocation to a specific subsystem.

For example:

```
//DD1 DD DSN=CSC.DATASET,DISP=(NEW,KEEP),UNIT=3490
```

Assume the following:

- The unit name 3490 contains some drives owned by MVS/CSC and others owned by the HSC.
- The TAPEREQ below has been defined to the HSC.

  ```
  TAPEREQ DSN(CSC.DATASET) ESOTERIC(CSCDRVS) RECT(36B)
  ```

- The esoteric CSCDRVS contains all 3490 drives, and all are owned by the CSC.

The CSC owns the allocation request because it owns all the drives in esoteric CSCDRVS.

Subsystem User Exits 02 and 08

The SMC evaluates the HSC and MVS/CSC user exit policies to resolve allocation request ownership. If an MVS/CSC user exit requests a specific MVS/CSC, the requested MVS/CSC is always considered the request owner. If the user exit requests a nonlibrary allocation, or requests allocation to an esoteric containing no drives known to this subsystem, the SMC assumes that another subsystem may be the appropriate owner for the request. Other user exit return codes imply that this subsystem is the owner of the request.

See Appendix A, "JES2 User Exit Return Codes and Library Subsystem Ownership" for details about how user exit return codes determine the request owner.

Specific Volume Location

The SMC queries each active library subsystem to determine whether the volume resides in its library. The first subsystem that locates the volume owns the allocation request.

Scratch Volume Availability

The SMC queries each active library subsystem to determine the availability of the requested media, subpool, and label type. The first subsystem containing eligible scratch volumes owns the allocation request.

Note: The SMC performs identically to prior NCS allocation processing to determine availability of scratch volumes. Any subpool not defined to the library subsystem is treated as the general subpool (subpool 0). This subpool returns a "volume available" status if it contains scratch volumes.

SMC DFSMS Processing

The SMC interface to DFSMS provides the following capabilities:

- esoteric substitution during MVS JCL interpretation
- at device allocation time, the DFSMS DATACLAS value can request a specific recording technique and/or media.
- at device allocation time, the DFSMS MGMTCLAS value can be used for VTV allocations.
- at volume mount, the DFSMS MGMTCLAS value can be assigned to a VTV.
- at volume mount, the DFSMS DATACLAS value can request a media for a scratch request.

Enabling/Disabling the SMC DFSMS Interface

To enable the SMC DFSMS interface, specify the SMS parameter of the ALLOCDef command as follows:

```
ALLOCDEF SMS=ON
```

To disable the SMC DFSMS interface, specify the SMS parameter of the ALLOCDef command as follows:

```
ALLOCDEF SMS=OFF
```

The SMC DFSMS interface can be selectively enabled or disabled by job or job step by specifying the SMS parameter of the ALLOCJob command.

DATACLAS, STORCLAS, and MGMTCLAS Specification

DFSMS DATACLAS can be specified by using the DATACLAS parameter on the DD JCL statement or by executing an Automatic Class Selection (ACS) routine. STORCLAS and MGMTCLAS can only be specified by ACS routines.

Note: STORCLAS and MGMTCLAS JCL parameters are not supported by the SMC DFSMS interface due to conflicts with IBM MVS DFSMS. Using the STORCLAS JCL parameter causes a data set to become DFSMS-managed, and the MGMTCLAS JCL parameter requires a DFSMS-managed data set. The data sets assigned STORCLAS and MGMTCLAS values in the STKTAP1 environment are not actually DFSMS-managed.

Invoking ACS Routines

IBM DFSMS invokes ACS routines with the variable &ACSENVIR set to ALLOC before the SMC invokes the ACS routines with variable &ACSENVIR set to STKTAP1.

The SMC invokes the ACS routines at the following points in processing:

- SSI55 Interpreter/Dynamic Allocation Exit (IDAX)
- SSI24 common allocation
- mount message interception.

The ACS routines are invoked in the following order:

1. data class
2. storage class
3. management class
4. storage group.

Management class and storage group ACS routines are called only if a storage class is assigned.

DFSMS Automatic Class Selection (ACS) Routine Environment for SMC

The following list of read-only variables is passed by SMC to DFSMS when the information is available to the SMC. Not all variables are available for every call to the ACS routines. See the descriptions of each DFSMS interface for exceptions.

- &ACSENVIR (equals STKTAP1 for the SMC interface)
- &ALLVOL
- &ANYVOL
- &DATACLAS
- &DD
- &DSORG
- &DSN
- &DSTYPE
- &EXPDT
- &FILENUM
- &JOB
- &LABEL
- &NVOL
- &PGM
- &RETPD
- &SYSNAME
- &SYSPLEX
- &UNIT.

In the STKTAP1 environment, the &ANYVOL variable is used only to match a specific VOLSER and does not contain the "REF=*xx*" values for VOL=REF allocations.

The &DATACLAS field is set when the JCL DD statement specifies this parameter.

Esoteric Substitution During IDAX

When a job is in the JCL interpretation phase, the unit name (esoteric) can be changed indiscriminately (for example, you could substitute disk for tape). Considerations:

- Esoteric substitution applies to all newly created data sets. New allocations are signified by specifying DISP=NEW, or in some cases, DISP=MOD in the JCL (refer to the IBM *MVS JCL Reference* manual for a list of requirements when using DISP=MOD to specify a new data set).
- Unit affinity chains are always separated if ACS routines return different esoterics for different members of the chain.
- VOL=REF chains within a job are validated and updated with the head-of-chain esoteric if necessary to ensure that volume references are honored.

When the SMC locates a new allocation that is not DFSMS-managed, the SMC calls the DFSMS ACS routines with the environment variable `&ACSENVIR` set to STKTAP1. When the ACS routines return a pair of storage group names with the first name being STK1, the second storage group name replaces the original value of the UNIT parameter of the DD statement. For example, the following DD statement allocates a 3490 nonlibrary tape drive (esoteric TNLIB):

```
//DD1  DD DSN=&TEMP.DSN,UNIT=TNLIB,DISP=NEW
```

Assume that your installation wants to move all new tape allocations into a tape library. You can code the DFSMS ACS routines to return storage groups STK1 and a library esoteric such as TACS0, where TACS0 contains all drives in library location ACS0. After esoteric replacement, the only drives considered for the allocation are contained within the TACS0 esoteric. The nonlibrary drives are no longer considered eligible for the allocation.

Defining Storage Groups and Storage Classes

To enable esoteric substitution during IDAX, complete the following steps:

1. Define a storage group of type POOL named STK1.
2. Define a storage group of type POOL for each possible esoteric returned, giving it the same name as the esoteric name.
3. Define at least one volume to each storage group. StorageTek recommends that you use a nonexistent volume for this definition.
4. Create a storage class ACS routine that, when it is invoked by the SMC (i.e., when the environmental variable is `&ACSENVIR=STKTAP1`), returns a storage class to be passed to the storage group ACS routine. Refer to Figure 1 on page 38.

5. Create a storage group ACS routine that, when the environmental variable is `&ACSENVIR=STKTAP1`, returns two storage groups (e.g., `&STORGRP='STK1','CART'`). In this case, "`STK1`" is the first storage group returned, and "`CART`," the replacement esoteric, is the second. Refer to Figure 1.
6. Test for the `&ACSENVIR=STKTAP1` read-only variable in the storage class and storage group ACS routines. This enables you to prevent the storage class being assigned when MVS invokes the ACS routines. If MVS invokes the SMC version of the ACS routines, the data set becomes DFSMS-managed and all SMC allocation functions are bypassed.
7. Return both a storage class and a storage group during the SMC invocation of the DFSMS ACS routines. If a storage class is returned but a storage group is not, DFSMS issues a message stating that allocation has failed, but in reality that may not be the case.

```
PROC STORCLAS

FILTLIST LOCALDSN INCLUDE (BACKUP*.**,
                           PROD.BKP*.**)

FILTLIST RMTDSN INCLUDE (PROD.OFFSITE.**)

  IF &ACSENVIR = 'STKTAP1' THEN
    SELECT
    WHEN &DSN = &LOCALDSN
      SET &STORCLAS = 'CART'
    WHEN &DSN = &RMTDSN
      SET &STORCLAS = 'RMTCART'
    END
  END

================================================================================

PROC STORGRP

  IF &ACSENVIR = 'STKTAP1' THEN
    SELECT
    WHEN &STORCLAS = 'CART'
      SET &STORGRP = 'STK1','CART'
    WHEN &STORCLAS = 'RMTCART'
      SET &STORGRP = 'STK1','RMTCART'
    END
  END
```

Figure 1. Creating Storage Class/Storage Group Routines

Availability of Read-only Variables

During DFSMS STORCLAS/STORGRP ACS routine processing, all of the read-only variables listed in "DFSMS Automatic Class Selection (ACS) Routine Environment for SMC" on page 36, except `&DSORG`, are available to the IDAX interface for esoteric substitution. The `&DATACLAS` field is set when the DD statement in the JCL specifies this parameter.

Validating DFSMS STORCLAS/STORGRP ACS Routine Execution

DFSMS provides the following ways to validate correct ACS routine execution:

- DFSMS WRITE statement in the ACS routines
- DFSMS ISMF test function.

When the SMC invokes the DFSMS ACS routines, all messages issued by DFSMS WRITE statements are routed to the GTF trace file if SMC tracing is active.

See the publication *DFSMS/MVS DFSMSdfp Storage Administration Reference* for further information about writing and testing DFSMS ACS routines.

Retrieving DFSMS Values During Allocation and Volume Mount

Requesting Recording Technique/Media using DFSMS DATACLAS

During allocation processing, the media and/or recording technique can be specified for a DD allocation by coding:

- TAPEREQ statement
- DATACLAS parameter on the DD statement
- DATACLAS ACS routines.

During scratch volume mount processing, only the volume media can be specified.

Note: DATACLAS media and recording technique values override TAPEREQ values.

Media/Rectech Support

The following media and recording techniques are supported:

Media:	
MEDIA1	Standard capacity for both 18-track and 36-track media types (this is the default)
MEDIA2	Enhanced capacity 36-track media
Recording Techniques:	
18TRACK	18-track recording technique
36TRACK	36-track recording technique

Defining Data Classes

To use DATACLAS, customers must define their own data class and specify the recording technique and media. If the recording technique is left blank, the SMC defaults to allowing both 18TRACK and 36TRACK to remain eligible for the allocation request, unlike MVS, which defaults to 36TRACK.

As an example, a customer can define a DATACLAS named STD18 that specifies a recording technique of 18TRACK and standard media.

When using an ACS routine to supply a DATACLAS to set recording technique and media, do not specify `&ACSENVIR='STKTAP1'`. The DATACLAS ACS routine is invoked both by the SMC and MVS, and the resulting recording technique and media should be consistent.

Refer to Figure 2 on page 40 for an example of how to define a data class routine.

```
PROC DATACLAS

FILTLIST RMTDSN INCLUDE(PROD.OFFSITE.**)

  SELECT
  WHEN &DSN = &RMTDSN
    SET &DATACLAS = 'ECART36'
  END
```

Figure 2. Creating Data Class Routines

In the example above, assume DATACLAS ECART36 is defined with a recording technique of 36-track and a media type of MEDIA2. This data class routine example assigns ECART36 to data sets with names that start with "PROD.OFFSITE."

By default, any DATACLAS value specified by the DFSMS ACS routine overrides the DATACLAS parameter specified on the JCL statement. Your installation can change this behavior by adding the following statements to your DFSMS DATACLAS ACS routine.

```
WHEN (&DATACLAS NE '')
DO
  SET &DATACLAS = &DATACLAS
EXIT
END
```

Requesting Management Class Using DFSMS ACS Routines

During allocation and mount processing, a management class name can be specified for a virtual allocation by coding any of the following:

- TAPEREQ statement
- MGMTCLAS ACS routine.

The MGMTCLAS JCL parameter is not supported by the SMC DFSMS interface.

Note: The MGMTCLAS management class name overrides a TAPEREQ management class name.

When writing a management class routine, keep in mind:

- The management class routine is invoked only when a storage class is assigned.
- The management class routine must test for the `&ACSENVIR='STKTAP1'` read-only variable value.
- During volume mount message IEC501A interception, the `&UNIT` read-only variable contains the generic unit type, such as 3490. Therefore, careful consideration should be taken when coding ACS routines that use the `&UNIT` read-only variable.

Refer to Figure 3 on page 42 for an example of how to define a management class routine.

```
PROC STORCLAS

FILTLIST CART INCLUDE ('CART')

  IF &ACSENVIR = 'STKTAP1' THEN
    SELECT
    WHEN (&UNIT EQ &CART)
      SET &STORCLAS = 'VIRTAPE'
    WHEN (&UNIT NE &CART)
      SET &STORCLAS = 'STKDFLT'
    END
  END

/*******************************************************************/

PROC STORGRP

  IF &ACSENVIR = 'STKTAP1' THEN
    SELECT
    WHEN (&STORCLAS = 'VIRTAPE')
      SET &STORGRP = 'STK1', 'VDRIVES'
    END
  END

/*******************************************************************/

PROC MGMTCLAS

FILTLIST LOCAL INCLUDE(BACKUP*.**.,
                       PROD.BKP*.**)

FILTLIST REMOTE INCLUDE(PROD.OFFSITE.**)

  IF &ACSENVIR = 'STKTAP1' THEN
    SELECT
    WHEN (&DSN = &LOCAL)
      SET &MGMTCLAS = 'INVTAPE'
    WHEN (&DSN = &REMOTE)
      SET &MGMTCLAS = 'OFFVTAPE'
    END
  END
```

Figure 3. Creating Management Class Routines

Note: In this example, the STORCLAS ACS routine assigns a storage class to every call. This ensures that the MGMTCLAS routine is also driven at mount time. Only the VTCS component of NCS currently uses the management class value.

Availability of Read-only Variables

- At allocation time and at mount time processing of message IEF233A, the `&UNIT` read-only variable is set by the SMC to the value specified by the `UNIT=` parameter of the DD statement. At mount time processing of message IEC501A, the SMC sets the `&UNIT` read-only variable to the generic unit type, such as 3490.
- During dynamic allocation, the `&DATACLAS` value can be specified by coding the DADACL text unit. However, this value is not available to the SMC when processing mount message IEF233A.

Validating DFSMS ACS Routine Execution

DFSMS provides the following ways to validate correct ACS routine execution:

- DFSMS WRITE statement in the ACS routines
- DFSMS ISMF test function.

When the SMC invokes the DFSMS ACS routines, all messages issued by DFSMS WRITE statements or by DFSMS processing are routed to the SYSMSG data set of the job and are also routed to the GTF trace file if SMC tracing is active.

See the publication *DFSMS/MVS DFSMSdfp Storage Administration Reference* for further information about writing and testing DFSMS ACS routines.

Interaction with Other Software

Open Type J

The SMC allocation enhancements may not operate if you use the MVS Open Type J macro. Since that macro allows you to change volume serial numbers or data set names at open time, information available at Job Step Allocation time may be incorrect as interpreted by the SMC.

Note: Some vendor software products use MVS Open Type J. If you are experiencing unexpected allocation results using a vendor software product, check with the vendor to determine if Open Type J is used and follow the recommendations below.

SMC allocation may be influencing MVS allocation erroneously based on information that may have changed at open time. To prevent this problem when using the Open Type J macro, specify the appropriate esoteric in the JCL or specify an appropriate esoteric in an applicable TAPEREQ control statement.

SAMS: DISK (DMS)

Sterling Software's SAMS: DISK (DMS) has two methods of allocating transports:

- allocates transports at session start-up, holds onto the transports throughout the session and uses Open Type J (refer to the section about ''Open Type J'').
- uses dynamic allocation (DYNALLOC) to allocate transports when required.

The SMC allocates correctly when dynamic allocation is used. Therefore, the latter method of allocating transports is recommended.

CA-MIA Tape Sharing

The Computer Associates Unicenter CA-MIA Tape Sharing for the z/OS and OS/390 product relies upon direct modification of the EDL at SSI24 time to determine what tape drives remain eligible for an allocation event. However, the SMC does not directly modify the EDL as part of its normal allocation processing. To enable proper coexistence with CA-MIA Tape Sharing, alter the ALLOCDef command to turn MIAcompat(ON).

Differences Between SMC and NCS Allocation

Drive Exclusion

The SMC influences allocation incrementally based on an ordered list of exclusion levels. The SMC allows a job to fail when not enough drives remain eligible after performing the minimum exclusion level. The HSC and MVS/CSC do not allow a job to fail before initiation, but the job may fail at execution time due to incompatible drive selection.

Recording Technique Exclusion

The SMC uses the requested recording technique as the last criterion in drive exclusion (exclusion level 8). If your installation has been using TAPEREQ or SMS recording technique to control allocation, such as specifying RECTECH(18TRACK) to direct allocation outside the library, you may need to change your TAPEREQ parameters. For example, if the library contains available scratches, all nonlibrary drives are excluded at exclusion level 6, and the allocation goes inside the library.

Note: StorageTek recommends that TAPEREQ statements specifying an 18-track or 36-track recording technique be changed to use an esoteric instead. Since the SMC allows any valid esoteric substitution, specifying TAPEREQ ESOTERIC(3480), which is applied at exclusion level 4, accomplishes the goal of directing the allocation to a 3480 device.

Bypassing Jobs

The SMC can bypass allocation influence for individual jobs if requested. The SMC ALLOCJob command BYPASS parameter allows a job to run as if no SMC and supporting HSC or MVS/CSC subsystem(s) are present.

Unit Affinity

The HSC ALLOC command and MVS/CSC ALTER command parameters GDGALL and UNITAFF are no longer supported and have been replaced by the SMC ALLOCDef and ALLOCJob command SEPLVL parameter during drive exclusion processing.

SEPLVL applies to both unit affinity and GDGALL chains. The default value for SEPLVL, MAX, always separates chains if no common eligible drives remain after drive exclusion has processed each member of the chain for all exclusion levels.

If your installation previously operated with GDGALL and UNITAFF set to NOSEP and would like the SMC to operate similarly, specify:

```
ALLOCDEF SEPLVL(3)
```

If your installation previously operated with UNITAFF and GDGALL set to SEP, then the default SEPLVL value (MAX) provides comparable functionality.

User Exit Parameters

The SMC does not honor the following HSC ALLOC or MVS/CSC ALTER command parameters:

- UX02SUB
- UX08SUB.

The SMC operates the same as the HSC and MVS/CSC did when Honor was specified for the UX02SUB and UX08SUB parameters of the HSC ALLOC command or the X02sub and X08sub MVS/CSC startup parameters.

User Exit Processing

In SMC allocation, User Exit 10 is called only once for each original chain. User Exit 08 (Specific Volume Exit) is called for each specific volume in the chain and User Exit 02 (Scratch Volume Exit) is called for each scratch request in the chain. Refer to "Affinity Separation" on page 27 for more information about unit affinity chain processing.

Esoteric Unit Name Substitution

The SMC does not require a TAPEREQ or user exit-supplied esoteric (from a TAPEREQ or user exit return value) to be a subset of the original EDL, nor does it require the esoteric name to be a library subsystem-defined or virtual esoteric unit name. Any valid esoteric unit name may be specified. The SMC uses the specified name to exclude drives not found in the current list of eligible drives.

DFSMS IDAX Esoteric Substitution

The SMC DFSMS IDAX interface call to the ACS routines (`&ACSENVIR=STKTAP1`) follows the IBM call to the ACS routines (`&ACSENVIR=ALLOC`).

Unit affinity chains are always separated when members of the chain receive different esoteric values.

Volume reference chains always propagate the esoteric received for the first use of the volume throughout the chain.

Messages

The MSGDEF or MSGJOB command can suppress SMC messages by severity level.

HSC message SLS1350I and MVS/CSC message SCS1350I are replaced with SMC messages SMC0045 and SMC0046. These messages display conflicting allocation criteria.

Chapter 4. SMC Allocation in a JES3 Environment

Overview

This chapter describes SMC allocation in a JES3 with SETUP environment. The following topics are discussed:

- SMC allocation – JES3 not managing drives
- SMC allocation – JES3 managing drives
- DFSMS esoteric substitution
- JES3 initialization parameter considerations
- SMC normal operations
- JES3 constraints
- multiple library subsystem support
- SMC DFSMS processing
- interaction with other software
- differences between SMC and NCS allocation.

SMC Allocation - JES3 Not Managing Drives

If JES3 is not managing any devices and SETUP=NONE has been specified on the JES3 STANDARDS initialization statement, the SMC operates the same as it does in a JES2 environment.

If JES3 is not managing any cartridge drives but is managing other types of devices, specify the J3NOSET parameter on the EXEC statement of the SMC START procedure (refer to "Creating the SMC START Procedure" on page 4). When J3NOSET is specified, the SMC operates the same as it does in a JES2 environment.

If either SETUP=NONE or J3NOSET is specified, no Type 1 modifications need to be installed on your JES3 system. HSC and MVS/CSC user exits invoked are the JES2 versions. For example, in HSC, the specific volume user exit SLSUX08 is invoked rather than user exit SLSUX13.

See Chapter 3, "SMC Allocation in a JES2 Environment" section for all further information about how the SMC operates in a JES2 environment.

SMC Allocation - JES3 Managing Drives

The SMC supports JES3-managed drives. JES3 manages drives through SETUP processing, which allocates drives identified on SETNAME statements when JOB, HWS (high watermark setup), or THWS (tape high watermark setup) is specified on the SETUP parameter of the JES3 STANDARDS initialization statement. In this environment, JES3 must manage all cartridge drives for the SMC to operate correctly.

SMC support operates during the following MVS subsystem interfaces (SSIs) and JES3 component phases:

- SSI55 Interpreter/Dynamic Allocation Exit (IDAX)
- JES3 Converter/Interpreter (C/I)
- SSI23 JES3 Dynamic Allocation
- JES3 Main Device Scheduler (MDS)
- SSI24 common allocation.

SSI55 Interpreter/Dynamic Allocation Exit (IDAX)

During MVS JCL interpretation processing, IDAX provides an opportunity to examine new data set allocations and to replace the unit parameter value with a different value (esoteric).

Note: This function is optional for SMC customers. For information about implementing DFSMS esoteric substitution, refer to "SMC DFSMS Processing" on page 90.

JES3 Converter/Interpreter (C/I)

During JES3 C/I subtask SWA processing, the SMC retrieves the media and recording technique obtained from the DFSMS DATACLAS specified by a DATACLAS JCL parameter on the DD allocation. This support is optional. For information about implementing DFSMS media and recording technique support, refer to "SMC DFSMS Processing" on page 90.

During JES3 C/I POSTSCAN processing, the SMC substitutes an esoteric to eliminate unacceptable drives from the allocation. The SMC performs the following processes to arrive at the best set of eligible drives:

- drive exclusion
- affinity separation
- esoteric unit name replacement.

At the end of JES3 C/I POSTSCAN processing, the SMC can defer the allocation until the job enters the initiator according to the customer's library subsystem defer policy. Also, at this point of processing, fetch messages can be suppressed according to the customer's library subsystem fetch policy.

These processes are discussed in this chapter.

SSI23 JES3 Dynamic Allocation

During SSI23 JES3 Dynamic Allocation processing, the SMC performs the same functions for dynamic allocations that the POSTSCAN C/I processes for common allocations:

- drive exclusion
- GDGALL affinity separation
- esoteric unit name replacement
- mount deferral.

JES3 Main Device Scheduler (MDS)

At the beginning of JES3 MDS processing, the SMC provides the ability to suppress fetch messages for dynamic allocation requests according to the customer's library subsystem fetch policy.

During MDS device selection, the SMC sets preference values for drives according to their relative desirability, that is, JES3 selects the available drive with the highest preference value for the allocation.

SSI24 Common Allocation

If a mount has been deferred until the job enters an initiator, during SSI24 common allocation processing, the mount may be deferred further until the data set is opened. The customer's library subsystem defer policy determines whether or not the mount is deferred.

Exceptions

The SMC does not influence the following types of cartridge tape allocation:

- demand allocation (i.e., request for a specific drive(s))

Notes: The SMC does perform DEFER processing for demand allocation.

- allocations excluded explicitly by entering the ALLOCJob command BYPASS parameter. Refer to the description of this command in Chapter 6, “Operator Commands”.
- allocations where the list of eligible devices does not contain any drives known to a library subsystem, such as drives contained in other vendor's tape libraries
- DFSMS-managed allocation. An SMS-managed data set is defined as a data set that has a storage class defined. A storage class is assigned when either
  - the STORCLAS parameter is specified on the DD statement, or
  - an installation-written ACS routine selects a storage class for a new data set.

DFSMS Esoteric Substitution

During MVS JCL interpretation processing, the Interpreter/Dynamic Allocation Exit (IDAX) provides an opportunity to examine new data set allocations and to replace the unit parameter value with a different value (esoteric).

This function is optional for SMC customers. For information about implementing DFSMS esoteric substitution, refer to “SMC DFSMS Processing” on page 90.

Drive Exclusion

The JES3 C/I process creates an Intermediate Job Summary Table (IJS) that represents the device requirements for the JES3 managed devices for an entire job. The Type 1 modification to JES3 C/I module IATIIP1 enables the SMC to examine the IJS and to determine whether the SMC should influence the allocation.

When the SMC determines an interest in the allocation, its first step is to narrow down the list of eligible drives to those that best support the allocation request. This step is called *drive exclusion*.

Like the HSC and MVS/CSC device separation processes, the SMC drive exclusion process does **not** take into account the status of drives (e.g., offline, busy) when selecting drives that are eligible for the allocation request. If all drives compatible with the allocation request are unavailable, the job goes onto the allocation wait queue.

The SMC performs drive exclusion by building a list of drives defined to the original unit name and then by following a step-by-step process using an ordered list of exclusion criteria to remove drives from this list.

Before any drives are removed from consideration for the allocation, the SMC first determines whether a job can be allocated.

For example:

```
//JOB1  JOB …
//STEP1 EXEC PGM=…
//DD1   DD   UNIT=349Ø,…
//DD2   DD   UNIT=349Ø,…
        .
        .
        .
//DD25  DD   UNIT=349Ø,…
```

Assume STEP1 is requesting twenty-five 3490 drives. If JES3 has configured only twenty-four 3490 drives, JES3 cannot allocate the required drives for the job. In this instance, two messages are issued:

- SMC0042, which indicates the job cannot allocate before SMC modification and allows the job to fail allocation.
- IAT4801, a JES3 message which indicates the job has been express-canceled by the interpreter Dynamic Support Program (DSP).

Once the SMC determines a job can be allocated, the SMC begins excluding drives based on an ordered list of criteria. Table 3 on page 55 and Table 4 on page 60 list these criteria and show the order the exclusion process follows. The lower the criteria level, the higher its importance. Thus, level 2 is considered more important than level 6.

Minimum Exclusion Level

By default, level 2 is the *minimum exclusion level*, that is, the desired minimum level of allocation exclusion, for both specific and scratch volume processing. However, if necessary, the ALLOCDef and ALLOCJob commands can reset the minimum exclusion level (refer to Chapter 6, "Operator Commands" for descriptions of these commands).

Note: If your VTCS configuration allows recalls of all migrated virtual volumes to any VTSS, StorageTek recommends you set the minimum exclusion level to 1 using the ALLOCDef command. See "Exclusion Level 2" on page 56 for more information.

In the first step of drive exclusion, the SMC determines if the job can be allocated at the minimum drive exclusion level. If not enough drives remain eligible at the minimum exclusion level, the SMC sets a JES3 indicator that causes JES3 to express-cancel the job.

For example:

```
//JOB2  JOB …
//STEP1 EXEC
//DD1   DD   UNIT=3490,VOL=SER=984001,DISP=OLD
//DD2   DD   UNIT=3490,VOL=SER=984002,DISP=OLD
//DD3   DD   UNIT=3490,VOL=SER=984003,DISP=OLD
```

In this system environment, imagine that twenty 3490-type drives have been configured. Only two of those twenty are actually 9840 drives. If all three volumes require a separate 9840 tape drive, then not enough 9840 drives exist for the job to allocate. Without SMC influence, JES3 would allow the job to proceed through allocation. However, with SMC influence, the job fails and two messages are issued:

- SMC0043, which indicates the job could not allocate at the specified minimum exclusion level.
- IAT4801, a JES3 message which indicates the job has been express-canceled by the interpreter DSP.

Note: No exclusion criteria (including minimum level) are applied when the user specifies demand allocation (i.e., requests a specific drive or drives).

Subsequent Levels

Once the minimum level of exclusion has succeeded, the remaining exclusion levels are performed. Any level above the minimum level that fails is skipped and exclusion continues to the next level.

Note: A level fails if it results in too few drives for the DD statement to allocate. The only required exclusion levels are those at the minimum level or below.

After the last drive exclusion level is completed, the SMC examines the final drive lists. If the job cannot be allocated because not enough drives remain eligible, drive exclusion regresses to the previous exclusion level for the failing DD statement. This process repeats until the job can allocate.

Intentionally Failing a Job

Depending upon the characteristics of your installation, you may want specific jobs to fail at allocation time rather than execute them with non-optimal devices. As an example, if you have two ACSs, one of which is located at a remote site, you may prefer to fail the job, rather than having to transfer the correct volume from one ACS to another.

For example:

```
//DD1  DD  UNIT=CART,VOL=SER=INACSØ,DISP=OLD
//DD2  DD  UNIT=AFF=DD1,VOL=SER=INACS1,DISP=OLD
```

Assume the following:

- Both volume INACS0 and INACS1 share common media characteristics.
- Volume INACS0 resides in ACS0, and volume INACS1 resides in ACS1.
- Only ACS0 contains a compatible drive.

If this affinity chain is not broken, then volume INACS1 must be ejected from ACS1 and entered into ACS0 for this job to execute, which may not be possible given the geographical locations of the ACSs.

In this case, failure at allocation time may be preferable to failure at run time. By specifying the correct MINLVL value (i.e., by issuing the ALLOCJob command with a MINLVL of 7 for this job), the job is failed at allocation time.

Caution: Setting the minimum exclusion level may cause jobs to fail unexpectedly. Whenever the SMC is unable to apply an exclusion criterion at or below the minimum level, the job fails allocation.

For example, if a TAPEREQ specifies a media of STANDARD for a scratch volume and the eligible device list contains only 9840 drives, when the SMC attempts to apply the “policy media” exclusion level, all drives are excluded. If the minimum level is set to 3 or higher, the job fails allocation. Therefore, StorageTek recommends that the default exclusion level (as specified by the ALLOCDef command) should normally not be set to a level greater than 2.

Specific Volume Request Drive Exclusion Criteria

For a specific volume allocation, the SMC excludes drives in order from lowest exclusion level to highest based on the criteria in Table 3. The lower the level number, the more important the exclusion criteria.

Table 3. JES3 Drive Exclusion Levels

Level	Specific Volume Criteria	Keyword*
1	Exclude drives incompatible with the volume media. Primary source: external volume label Secondary source: VOLATTR MEDIA parameter	MEDRECTECH
2**	For virtual volumes only, exclude drives not located in an accessible VTSS. Note: No further exclusion is performed for virtual volumes.	VTSSLOCATION
3	Exclude drives based on the required recording technique. Source: VOLATTR RECTECH parameter	VOLATTRRECTECH
4	Exclude drives based on user location policies. Primary source: TAPEREQ ESOTERIC parameter. Secondary source: HSC or MVS/CSC User Exit 12 or 13 return codes.	USERPOLICY
5	Exclude drives based on the HSC ALLOC command SPECVOL parameter. Note: This exclusion level does not apply to MVS/CSC.	SPECVOL
6	Exclude drives based on volume location type, i.e., library or nonlibrary.	LOCTYPE
7	Exclude drives based on the ACS location of the volume, if the volume resides in the library.	ACSLOCATION
8	Exclude drives based on the requested recording technique. Primary source: DFSMS data class recording technique. Secondary source: TAPEREQ RECTECH parameter.	POLRECTECH

* Keywords associated with each exclusion level are specified in the exclusion criteria displayed in message SMC0046.

** Level 2 is the default minimum level.

Exclusion Level 1

The SMC excludes drives that are not compatible with the volume media. The volume media can be obtained from the volume label or from an HSC VOLATTR statement MEDIA parameter.

Exclusion Level 2

If the specific volume requested is a virtual volume, the VTSS location for a resident volume must match the volume location. The VTSS location for a migrated virtual volume must match the location to which the volume can be recalled. The remaining exclusion levels do not apply to virtual volumes and are bypassed.

Note: If all VTSSs in your virtual configuration allow migration to common ACSs, you may want to set the minimum level exclusion level to 1 to allow jobs to proceed by migrating and recalling resident VTVs.

See the VTCS documentation for more information about virtual volumes.

Exclusion Level 3

If an HSC VOLATTR statement specified the recording technique (RECTECH parameter) for this specific volume, the SMC excludes drives that do not provide that recording technique.

For example:

```
//DD1 DD UNIT=TLACS0,VOL=SER=VOL000,DISP=OLD
```

A VOLATTR statement has been defined to the HSC.

```
VOLATTR SERIAL(VOL000) RECTECH(36B) MEDIA(ECART)
```

VOL000 requires a 9490 drive, and if TLACS0 contains both 9490 and 9490EE drives, the 9490EE drives are excluded from consideration for this allocation.

Note: In MVS/CSC, the VOLATTR RECTECH is implied by the composite recording technique of the drives in the list returned by the library server.

Exclusion Level 4

If a TAPEREQ statement specifies an esoteric unit name for this allocation request, the SMC verifies the esoteric unit name is defined to JES3. If the esoteric passes this test, any drives in the original unit name that are not also defined to the esoteric unit name are excluded from consideration for this allocation.

Note: The SMC does not substitute the requested esoteric directly, but uses it to alter the list of eligible drives.

For example:

```
//DD2   DD UNIT=CART,VOL=SER=TR0001,DSN=SYS4.TR1.DATA,DISP=OLD
```

The following TAPEREQ statement is defined to HSC or MVS/CSC.

```
TAPEREQ DSN(SYS4.TR1.**) ESOT(TLACS1) RECT(LONGI)
```

The drives that are not defined to the esoteric TLACS1 are excluded from consideration for this allocation.

If no TAPEREQ statement specifies an esoteric for this allocation request, HSC and MVS/CSC User Exit 12 or 13 return codes and values can define the criteria for drive exclusion at this level. The user exit return codes can direct the SMC to select library drives, nonlibrary drives, or a specific ACS, or the user exit can return an esoteric name that contains customer-selected drives. Any drives in the current list of eligible devices that did not match the user exit return values are excluded here.

For example:

```
//DD1    DD UNIT=CART,VOL=SER=VOL001,DISP=OLD
```

HSC User Exit 13 returns UX13LIB, which requests a library drive. The drives that are not in the library are excluded from consideration for this allocation.

Exclusion Level 5

The SPECVOL parameter of the HSC ALLOC command instructs the SMC which library ACS locations can be used for this request if one of two circumstances exists:

- the specific volume is not in the library and no drives are defined in the nonlibrary esoteric to the HSC, or
- the specific volume is not in the library, and user exit 13 requested a library location.

Drives not in SPECVOL ACS locations are excluded from consideration.

Note: This exclusion level does not apply to MVS/CSC.

For example:

```
//DD1   DD UNIT=CART,VOL=SER=NL0001,DISP=OLD
```

Volume NL0001 is not in the library and the HSC command ALLOC SPECVOL(00) has been issued.

If the NNLBDRV was not specified on the HSC LIBGEN SLILBRY macro meaning no drives exist outside the library, the drives not defined to ACS00 are excluded from consideration for this allocation.

Another example:

```
//DD2   DD UNIT=CART,VOL=SER=NL0002,DISP=OLD
```

Assume volume NL0002 is not in the library, and the HSC command ALLOC SPECVOL(01) has been issued to allow ACS01 to be used for specific nonlibrary volumes.

If nonlibrary drives were defined on the HSC LIBGEN SLILBRY macro but HSC user exit 13 returned UX13LIB, all nonlibrary drives and drives in ACS00 are excluded from consideration for this allocation.

Exclusion Level 6

The generic location of the volume (library or nonlibrary) reduces the remaining list of eligible drives.

Note: For exclusion level 6, nonlibrary drives are outside the library and contain **either** known (from the HSC NNLBDRV or MVS/CSC NONLIB esoteric) or unknown device characteristics.

If the volume resides in a library, all drives outside the library are excluded, and if the volume resides outside the library, all library drives are excluded.

For example:

```
//DD1   DD UNIT=CART,VOL=SER=LIB001,DISP=OLD
```

Assume volume LIB001 resides in a library.

All drives outside the library are excluded from consideration for this allocation.

Exclusion Level 7

For volumes residing in the library, the ACS location of the volume reduces the remaining list of eligible drives. Any drives that remain that do not reside in the same ACS as the volume are excluded.

For example:

```
//DD1   DD UNIT=CART,VOL=SER=A00VL1,DISP=OLD
```

Assume volume A00VL1 resides in ACS00, and the drives reside in ACS00 and ACS01.

The drives residing in ACS01 are excluded from consideration for this allocation.

Exclusion Level 8

Exclusion levels 1 and 3 restrict the list of eligible drives to those compatible with the volume's actual media and its recording technique, if specified on the HSC or MVS/CSC server VOLATTR statement. Exclusion level 8 may further restrict the drives for the request based on the TAPEREQ or DFSMS data class recording technique.

The HSC or MVS/CSC TAPEREQ definitions can explicitly denote a recording technique for the allocation request. DFSMS data class definitions can also specify an 18- or 36-track recording technique for the request and override any TAPEREQ recording technique specification.

For example:

```
//DD1   DD UNIT=CART,VOL=SER=TV9840,DISP=NEW,
//         DSN=SYS4.TR4.DATA
```

Assume the requested volume is a 9840 volume in the library with no recording technique specified on a VOLATTR. Assume an HSC TAPEREQ has been defined:

```
TAPEREQ DSN(SYS4.TR4.**) RECT(STK1RB)
```

Previous exclusion levels restricted the list of devices to all 9840 models. In this example, the TAPEREQ recording technique of STK1RB leaves only 9840B devices in consideration for this allocation.

Scratch Volume Request Drive Exclusion Criteria

For a scratch volume allocation, the SMC excludes drives in order from lowest exclusion level to highest based on the criteria in Table 4. The lower the level number, the more important the exclusion criteria.

Table 4. JES3 Drive Exclusion Levels

Level	Scratch Volume Criteria	Keyword*
1	For nonlabeled (NL) scratch volume requests, exclude all virtual drives.	VIRTUALLABEL
2**	No level 2 for scratch volume exclusion.	
3	Exclude drives based on the requested media. Primary source: DFSMS data class media specification. Secondary source: TAPEREQ MEDIA parameter. Tertiary source: HSC or MVS/CSC User Exit 04 request for virtual media.	POLMEDIA
4	Exclude drives based on user location policies. Primary source: TAPEREQ ESOTERIC parameter. Secondary source: HSC and MVS/CSC User Exit 04 and 12 return codes.	USERPOLICY
5	Exclude drives based on the media of available scratch volumes in subpool. Primary source: TAPEREQ SUBPOOL parameter. Secondary source: HSC or MVS/CSC User Exit 04 subpool values. Tertiary source: scratch subpool 0.	SUBPOOL
6	Exclude drives based on available scratch location type, i.e., library, nonlibrary, or virtual.	LOCTYPE
7	Exclude drives based on the HSC ALLOC or MVS/CSC ALTER command ZEROSCR parameter.	ZEROSCRATCH
8	Exclude drives based on the requested recording technique. Primary source: DFSMS data class recording technique. Secondary source: TAPEREQ RECTECH parameter.	POLRECTECH

* Keywords associated with each exclusion level are specified in the exclusion criteria displayed in message SMC0046.

** The MINLVL parameter value default, level 2 on the ALLOCDef or ALLOCJob command, applies to both scratch and specific volume requests. Even though level 2 has no meaning for scratch, level 2 is considered the default minimum level.

Exclusion Level 1

The SMC excludes virtual drives if the volume label type requested is NL (nonlabeled). This is the only required criterion. Refer to the VTCS documentation for more information about virtual volumes.

Exclusion Level 2

No level 2 scratch volume exclusion is performed.

Exclusion Level 3

Under HSC and MVS/CSC, a DFSMS data class definition can explicitly request the media desired, or the media can be derived from the DFSMS data class recording technique. Drives that do not provide the requested media support are excluded at this level.

If no DFSMS data class definition applies for this allocation, the HSC or MVS/CSC TAPEREQ definitions can explicitly specify the media desired for the allocation request, or media can be derived from TAPEREQ RECTECH. Drives that do not provide the requested media support are excluded at this level.

For example:

```
//DD1    DD UNIT=CART,DISP=(NEW,KEEP),DSN=SYS4.VIRT.DATA
```

No HSC TAPEREQ has been defined and no DFSMS data class definition applies. Assume HSC User Exit 04 return code requested virtual media (UX04VIRT). The HSC User Exit 04 media selected for this request is virtual. Only virtual drives remain in consideration for this allocation. The next example shows how TAPEREQ can override User Exit 04 return codes.

For example:

```
//DD1   DD UNIT=CART,DISP=(NEW,KEEP),DSN=SYS4.TR4.DATA
```

An HSC TAPEREQ has been defined:

```
TAPEREQ DSN(SYS4.TR4.**) MEDIA(DD3A) RECT(DD3)
```

Assume HSC User Exit 04 return code requested virtual media (UX04VIRT).

If no DFSMS data class definition or TAPEREQ definition applies to this allocation request, HSC or MVS/CSC User Exit 04 return code, UX04VIRT, can explicitly request virtual media. Non-virtual drives are excluded at this level.

However, the TAPEREQ media definition overrides the user exit return code, and the media selected for this request is helical. Only helical drives remain in consideration for this allocation. Messages SMC0045 and SMC0046 are issued indicating that the user exit request for virtual volumes was not honored.

Exclusion Level 4

If a TAPEREQ statement specifies an esoteric unit name for this allocation request, the SMC verifies the esoteric unit name is defined to JES3. If the esoteric passes this test, any drives in the original unit name that are not also defined to the esoteric unit name are excluded from consideration for this allocation.

Note: The SMC does not substitute the requested esoteric directly, but uses it to alter the list of eligible drives.

If no TAPEREQ statement specifies an esoteric for this allocation request, HSC or MVS/CSC User Exit 04 and User Exit 12 return codes and values can define this criterion for drive exclusion. The user exit return codes can direct the SMC to select library drives, nonlibrary drives, or a specific ACS, or the user exit can return an esoteric unit name that contains customer-selected drives. Any drives in the current list of eligible drives that did not match the user exit return values are excluded here.

For example:

```
//DD1    DD UNIT=CART,DSN=&&TEMP1,DISP=(NEW,PASS)
```

Assume HSC User Exit 04 returns UX04LIB, which requests library drives.

The drives that are not in the library are excluded from consideration for this allocation.

A TAPEREQ example is:

```
//DD2   DD UNIT=CART,DSN=SYS4.TR1.DATA,DISP=(NEW,KEEP)
```

The following TAPEREQ statement is defined to the HSC or MVS/CSC.

```
TAPEREQ DSN(SYS4.TR1.**) ESOT(TLIB9840) RECT(LONGI)
```

The drives that are not defined to the esoteric TLIB9840 are excluded from consideration for this allocation. Any User Exit 04 return codes would be ignored.

Exclusion Level 5

The combined media and specified recording technique of the volumes in a scratch subpool define this criterion for drive exclusion. An HSC or MVS/CSC TAPEREQ SUBPOOL parameter can specify a scratch subpool name for the request.

If no TAPEREQ SUBPOOL parameter is specified for this allocation, HSC or MVS/CSC User Exit 04 can return a scratch subpool number or subpool name when it also sets the return code to use default processing or library drives. HSC or MVS/CSC User Exit 12 can also apply to affinity chains here. Refer to “Affinity Separation” on page 66 for a discussion about the interaction between User Exit 04 and User Exit 12.

Note: When no specific subpool applies to the allocation, the default subpool, subpool 0, is used.

Drives that do not provide a recording technique compatible with a volume(s) in the subpool are excluded from consideration for this allocation.

For example:

```
//DD1    DD UNIT=CART,DSN=NEW.TRK36.DATA,DISP=(NEW,KEEP)
```

Assume HSC has scratches of media types 36TRACK and DD3 in a subpool named SUBPOOL3, and a TAPEREQ entry exists:

```
TAPEREQ DSN(NEW.*) SUBPOOL(SUBPOOL3)
```

Drives providing 36-track recording and helical recording remain in consideration for this allocation.

A User Exit 04 example for this criteria:

```
//DD1   DD UNIT=CART,DSN=SYS4.MYGROUP.DATA,DISP=(NEW,KEEP)
```

Assume the following:

- HSC User Exit 04 returns UX04LIB
- HSC User Exit 04 returns MYSUBPL in UX04SNAM
- MYSUBPL has been defined to the HSC using a SCRPOOL statement in the SLSSYSxx PARMLIB member or SCRPDEF file
- HSC has scratch volumes defined in MYSUBPL scratch subpool.

The drives that are not in the library are excluded from consideration for this allocation during level 4 drive exclusion.

Exclusion Level 6

The generic location of available scratch volumes provides the next level of drive exclusion. If the library contains no scratch volumes, all library drives and virtual drives are excluded from selection, leaving only nonlibrary drives eligible.

If the request is for a virtual scratch volume, eligible virtual drives are selected based on the management class from DFSMS or TAPEREQ; VTSSs that cannot support the requested management class migration are excluded. Last, if the library contains scratch volumes and virtual volumes were not requested, all virtual drives and all nonlibrary drives are excluded, leaving only library drives eligible.

For example:

```
//DD1   DD UNIT=CART,DSN=MY.NEW.DATASET,DISP=(NEW,KEEP)
```

Assume there are no MVS/CSCs, the HSC is active and has no nonlibrary drives defined, and the HSC library contains no scratch volumes.

All HSC library drives are excluded from consideration for this allocation. When all drives are excluded during an exclusion level, and the level is not at or below minimum level, then all drives excluded at that level are restored to the list of eligible drives and processing continues at the next level of exclusion.

In this case, all the HSC drives that were excluded by this criterion are now restored to the list. Processing continues at level 7.

Exclusion Level 7

The HSC ALLOC command and MVS/CSC ALTER command or startup parameter, ZEROSCR, is supported for JES3. ZEROSCR(OFF) provides drive exclusion found in earlier NCS releases. Drives residing in the ACS with the largest number of available scratch volumes remain eligible for selection. All other drives are excluded from selection. ZEROSCR(ON) excludes drives residing in ACSs that do not contain any scratch volumes.

For example:

```
//DD1   DD UNIT=CART,DSN=MY.NEW.DATASET,DISP=(NEW,KEEP)
```

Assume that ZEROSCR(OFF) has been specified, and two ACSs have scratch counts as follows: ACS0 has 400 and ACS1 has 500.

Only drives in ACS1 remain eligible for the allocation. Drives in ACS0 have been excluded.

Another example is:

```
//DD2   DD UNIT=CART,DSN=MY.OTHER.DATASET,DISP=(NEW,KEEP)
```

Assume ZEROSCR(ON) has been specified, and three ACSs have scratch counts as follows: ACS0 has 400, ACS1 has 500, ACS2 has 0.

Drives in ACS0 and ACS1 remain eligible. ACS2 drives are excluded from consideration because ACS2 does not contain any scratch volumes.

Exclusion Level 8

Exclusion level 3 restricts the list of eligible drives to those compatible with the TAPEREQ or DFSMS requested media. Exclusion level 5 restricts the list of eligible drives to those compatible with the scratch media in the requested subpool. Exclusion level 8 may further restrict the devices for the request based on the TAPEREQ or DFSMS data class recording technique.

The HSC or MVS/CSC TAPEREQ definitions can explicitly specify a recording technique for the allocation request. DFSMS data class definitions can also denote an 18- or 36-track recording technique for the request and will override any TAPEREQ recording technique specification.

This criterion is the last to be applied to the drives that are eligible for the allocation. Any drive that does not provide the requested recording technique is excluded from consideration for the allocation.

For example:

```
//DD1   DD UNIT=CART,DISP=(NEW,KEEP),DSN=SYS4.TR4.DATA
```

Assume an HSC TAPEREQ has been defined:

```
TAPEREQ DSN(SYS4.TR4.**) RECT(STK1RB)
```

Previous exclusion levels restricted the list of devices to all 9840 models. In this example, the TAPEREQ recording technique STK1RB leaves only 9840B devices in consideration for this allocation.

Affinity Separation

Explicit unit affinity is an MVS facility that allows volumes associated with two separate JCL DD statements, or allocation requests, to be mounted serially on the same drive. A request for all generations of a GDG group (GDG ALL chain) can be considered as a GDGALL affinity.

Unlike HSC and MVS/CSC, SMC makes no distinction between these two types of affinity. When processing an affinity chain begins, the drive exclusion process examines each allocation in the chain separately up to and including the minimum exclusion level. The chain is always separated when the minimum exclusion level processing results in lists of eligible drives, for two or more members of the chain, that do not contain common drives.

For example:

```
//DD1   DD UNIT=CART,DSN=MY.HELICAL.DATASET,DISP=OLD
//DD2   DD UNIT=AFF=DD1,DSN=MY.LONGI.DATASET,DISP=OLD
```

DD1 specifies a data set on helical media and DD2 specifies a data set on longitudinal media. Drive exclusion level 1 for specific volumes creates a list of eligible drives for each DD according to volume media required. The two lists do not contain a common drive. As a result, DD1 and DD2 no longer represent one drive allocation but two separate allocation requests. At this point, the SMC breaks the affinity chain between them.

Affinity Head-Of-Chain

For SMC affinity chain processing, the “head” of the affinity chain containing only scratch or only specific volumes is the first DD statement in the chain. If an affinity chain contains both scratch and specific volumes, the first specific volume is the “head.”

User Policy Influence on Affinity Separation

After the minimum level of drive exclusion and affinity separation completes, user policy influences the remaining affinity separation decisions.

Further affinity chain processing decisions are based on the first value found between the following:

- HSC or MVS/CSC User Exit 12 return code separation decision
- ALLOCDef or ALLOCJob command SEPLvl parameter value.

User Exit 12

HSC or MVS/CSC User Exit 12 sends an entire affinity chain to the user for examination. The user can decide whether the affinity chain remains together or separates during exclusion processing after the minimum level.

If User Exit 12 returns UX12SEP, indicating the chain members can be separated, the SMC proceeds as if ALLOCDEF SEPLVL=MAX were specified.

If User Exit 12 returns UX12LIB, UX12NLIB, UX12REPL, or UX12SACS, the affinity chain remains together for the remainder of drive exclusion processing. The location requested (e.g., library drives if UX12LIB is returned) replaces any location value provided by HSC or MVS/CSC User Exits 04 or 13. This location information is used by drive exclusion level 4. Refer to Table 3 on page 55 and Table 4 on page 60 for drive exclusion level information.

For example:

```
//DD1   DD UNIT=CART,DSN=SYS4.DATASET1,VOL=SER=NOLIB1,DISP=OLD
//DD2   DD UNIT=AFF=DD1,DSN=SYS4.DATASET2,VOL=SER=INLIB2,DISP=OLD
```

Assume the following:

- DD1 requests a nonlibrary volume and DD2 requests a library volume.
- User Exit 13 returns a nonlibrary esoteric for DD1.
- User Exit 12 returns a no separate decision and a library location.

The affinity chain stays together. Drive exclusion ignores the esoteric returned by User Exit 13 for DD1 and excludes all but library drives requested by User Exit 12.

Any one of the four return values UX12LIB, UX12NLIB, UX12REPL, and UX12SACS from User Exit 12 causes the SMC to operate as if ALLOCDEF SEPLvl=MIN were specified.

Note: Any HSC or MVS/CSC User Exit 12 separation decision overrides the SEPLvl parameter of the ALLOCDEf or ALLOCJob commands.

SEPLvl Parameter Value

If the decision to separate or keep the affinity chain together still has not been made, further affinity chain processing depends on the SEPLvl parameter value of the ALLOCDef and ALLOCJob commands (refer to Chapter 6, “Operator Commands,” for descriptions of these commands and setting an affinity separation level).

The SEPLvl parameter operates in three distinct ways depending on the value specified. The SEPLvl MAX parameter allows drive exclusion to be performed to the maximum level for each allocation element in the affinity chain. At the end of the drive exclusion process, if two or more elements of the chain no longer have common drives, the affinity chain is separated.

For example:

```
//DD1   DD UNIT=349Ø,VOL=SER=INACSØ,DSN=MY.DSN1,DISP=OLD
//DD2   DD UNIT=AFF=DD1,VOL=SER=INACS1,DSN=MY.DSN2,DISP=OLD
```

Assume the following:

- Both volume INACS0 and INACS1 share common media characteristics.
- Volume INACS0 resides in ACS0, and volume INACS1 resides in ACS1.
- ALLOCDEF SEPLVL(MAX) has been specified.

During specific volume exclusion level 7, the list of eligible drives in DD1 are all located in ACS0 and the list of eligible drives in DD2 are all located in ACS1. No drives are common to both lists. This affinity chain separates.

Note: MAX is the default value for SEPLvl.

The SEPLvl parameter can be specified as MIN. The value MIN requests that any affinity chains that remain together after the minimum level of drive exclusion should not be separated during the remaining levels of drive exclusion processing.

Using the same example:

```
//DD1   DD UNIT=349Ø,VOL=SER=INACSØ,DSN=MY.DSN1,DISP=OLD
//DD2   DD UNIT=AFF=DD1,VOL=SER=INACS1,DSN=MY.DSN2,DISP=OLD
```

Once again, assume:

- Both volume INACS0 and INACS1 share common media characteristics.
- Volume INACS0 resides in ACS0, and volume INACS1 resides in ACS1.
- ALLOCDEF SEPLVL(MIN) has been specified.

During specific volume exclusion level 7, the list of eligible drives for DD1 is all located in ACS0. DD1 is considered the “head” of the chain and exclusion is performed to ACS0. Only drives in ACS0 remain eligible for the allocation, and the affinity chain remains together.

The SEPLvl parameter can also be set to a specific drive exclusion level. The level number indicates the highest drive exclusion level at which separation can occur. Using the previous example, if the user specifies

```
ALLOCDEF SEPLVL=6
```

The chain remains together through drive exclusion level 6 because each level produces a drive list containing common drives for both DD1 and DD2.

Another example:

```
//DD1  DD UNIT=349Ø,VOL=SER=INACSØ,DSN=SYS4.DATASET1,DISP=OLD
//DD2  DD UNIT=AFF=DD1,VOL=SER=INACS1,DSN=SYS4.DATASET2,DISP=OLD
//DD3  DD UNIT=AFF=DD1,VOL=SER=NLØØØ1,DSN=REAL.DATASET,DISP=OLD
```

Assume the following:

- ALLOCDEF SEPLVL=5 has been specified.
- Volume INACS0 is in ACS0 and VOLATTR MEDIA(ECART) RECTECH(36B) has been defined.
- Volume INACS1 is in ACS1 and VOLATTR MEDIA(ECART) RECTECH(36B) has been defined.
- Volume NL0001 is a nonlibrary volume and `VOLATTR MEDIA(DD3A)` has been defined.

At drive exclusion level 1, two chains result because of incompatible media (ECART and DD3A). Statements DD1 and DD2 are still chained together and continue through exclusion levels 2 through 5 because the exclusion criteria do not create disparate sets of eligible drives for the two DD statements.

No further separation of the DD1/DD2 chain can occur because the SEPLVL parameter has been set to level 5.

After the last drive exclusion level is completed for the job, the SMC examines the final drive lists. If the job cannot be allocated because not enough drives remain eligible, drive exclusion regresses to the previous exclusion level of the failing DD statement and a separated chain may be rejoined. This process repeats until the job can allocate.

Note: The SEPLvl parameter replaces the function of the HSC ALLOC and MVS/CSC ALTER command UNITAFF and GDGALL parameters. These parameters are no longer supported.

If your installation previously specified UNITAFF(NOSEP) and GDGALL(NOSEP), to preserve that behavior, specify a SEPLvl value of 3 on the ALLOCDef command. If your installation previously specified UNITAFF(SEP) and GDGALL(SEP), no SEPLVL parameter change is required, since the default SMC behavior separates affinity chains at the maximum separation level.

Refer to the ALLOCDef and ALLOCJob commands in Chapter 6, “Operator Commands,” for more information about setting an installation default minimum exclusion level and unit affinity separation level.

Esoteric Unit Name Replacement

After drive exclusion and affinity separation successfully complete, each allocation may have a new list of eligible drives. The search begins to find an esoteric containing that exact list of drives. The SMC replaces the original JCL unit name in the Intermediate Job Summary Table (IJS) with this new esoteric.

The search for the "perfect" esoteric begins with the original JCL unit name or the unit name from the catalog entry for that data set. For example, assume the data set being allocated has been cataloged with the unit name 3490. Table 5 lists all the "3490" drives in the system.

Table 5. 3490 Drive List

ACS0	ACS1	Nonlibrary Location
0A10: 9490	0C10: 9490	0E10: 9490
0B10: 9840	0C11: 9490	0E11: 9490

JES3 groups devices by XTYPE names and groups XTYPE names by esoterics. The DEVICE statements coded in the JES3 initialization parameters follow:

```
DEVICE,TYPE=TA33490,XTYPE=(ACS09490,CA),JNAME=CA10,
 JUNIT=(A10,MVS1,TAP,ON),XUNIT=(A10,MVS1,TAP,ON)

DEVICE,TYPE=TA33490,XTYPE=(ACS09840,CA),JNAME=CA11,
 JUNIT=(B10,MVS1,TAP,ON),XUNIT=(B10,MVS1,TAP,ON)

DEVICE,TYPE=TA33490,XTYPE=(ACS19490,CA),JNAME=CC10,
 JUNIT=(C10,MVS1,TAP,ON),XUNIT=(C10,MVS1,TAP,ON)

DEVICE,TYPE=TA33490,XTYPE=(ACS19490,CA),JNAME=CC11,
 JUNIT=(C11,MVS1,TAP,ON),XUNIT=(C11,MVS1,TAP,ON)

DEVICE,TYPE=TA33490,XTYPE=(NLIB9490,CA),JNAME=CE10,
 JUNIT=(E10,MVS1,TAP,ON),XUNIT=(E10,MVS1,TAP,ON)

DEVICE,TYPE=TA33490,XTYPE=(NLIB9490,CA),JNAME=CE11,
 JUNIT=(E11,MVS1,TAP,ON),XUNIT=(E11,MVS1,TAP,ON)
```

Each unique location and device type pair has a unique XTYPE name. For example, the 9490 drive in ACS0 has a unique XTYPE name because it is the only 9490 in that location. The two nonlibrary 9490 drives share an XTYPE name because they are the same device type in the same location. An XTYPE should always define one type of device.

XTYPE names are associated with esoteric unit names in the initialization parameters as shown here:

```
SETNAME,XTYPE=ACSØ949Ø,NAMES=(CART,349Ø,LIBDRVS,ACSØDRVS,AØ949Ø)
SETNAME,XTYPE=ACSØ984Ø,NAMES=(CART,349Ø,LIBDRVS,ACSØDRVS,AØ984Ø)
SETNAME,XTYPE=ACS1949Ø,NAMES=(CART,349Ø,LIBDRVS,ACS1DRVS,A1949Ø)
SETNAME,XTYPE=NLIB949Ø,NAMES=(CART,349Ø,NLIBDRVS,NL949Ø)
```

Assume that during drive exclusion processing, the SMC determined the volume specified for this allocation resides in ACS0 and requires a 9490 drive. The drive exclusion process eliminates groups of drives by XTYPE.

In the environment defined above, the following XTYPE groups are no longer eligible for the allocation:

- ACS09840 – excluded at level 1 because 9840 drives are incompatible with the volume media.
- NLIB9490 – excluded at level 6 because the volume is in the library and these drives are not.
- ACS19490 – excluded at level 7 because the volume is in ACS0 and these drives are in ACS1.

One XTYPE, ACS09490, remains eligible for allocation at the end of drive exclusion. SMC esoteric unit name replacement now searches the SETNAME definitions for an esoteric that only contains the XTYPE ACS09490. For this allocation, the SMC selects the esoteric A09490 because it contains only XTYPE ACS09490. The A09490 esoteric replaces the original unit name, 3490, in the Intermediate Job Summary (IJS) table for that job.

If the example required two drives for the allocation (e.g., UNIT=(3490,2)) and the first volume to be mounted resides in ACS0, the results of drive exclusion would be as follows:

- ACS09840 – excluded at level 1 because 9840 drives are incompatible with the volume media.
- NLIB9490 – excluded at level 6 because the volumes are in the library and these drives are not.
- Exclusion level 7 fails.

On entry to level 7, three drives remain, two drives defined to XTYPE ACS19490 and the other drive to XTYPE ACS09490. If XTYPE ACS19490 were excluded because of ACS location, only one drive would remain. This allocation requires two drives. Thus, exclusion level 7 does not exclude the drives in ACS1.

Two XTYPEs, ACS09490 and ACS19490, remain eligible for allocation at the end of drive exclusion. The SMC esoteric unit name replacement now determines that XTYPE ACS09490 cannot be used for the allocation.

IBM APAR OW38427 to JES3 introduced the restriction that multi-unit allocations use devices defined in the same XTYPEs. Since XTYPE ACS09490 only contains one drive, it cannot satisfy the allocation requirements. The SMC esoteric unit name replacement now searches the SETNAME definitions for an esoteric that only contains the XTYPE ACS19490. The A19490 esoteric replaces the original unit name, 3490, in the IJS for that job.

After the SMC has updated the IJS, JES3 C/I processing continues. JES3 creates a Job Summary Table (JST) from the IJS table and performs any high watermark setup (HWS) chaining. During HWS chaining, JES3 can also change the esoteric unit name in the JST after the SMC changes the esoteric. The HWSNAME initialization statements define which esoteric unit names are subsets of other esoteric unit names. This allows JES3 to reuse devices in following steps.

Refer to “JES3 Initialization Parameter Considerations” on page 76 for more information about setting up your installation’s JES3 DEVICE, SETNAME and HWSNAME statements.

Suppressing Fetch Messages

By the time JES3 C/I processing completes, the IJS becomes the JST that represents the job for the remainder of its existence. The JST reflects the esoteric substitutions made by the SMC and by JES3. The next stage for the job is the Main Device Scheduler (MDS).

At the beginning of MDS processing, JES3 begins preparing the job for allocation. Asking the operator to fetch volumes is an optional phase in MDS. JES3 issues a fetch message when a job requires a volume that is not currently mounted and the SETPARAM statement FETCH parameter is set to YES (the default). If the SETPARAM statement also specifies ALLOCATE=MANUAL, jobs are placed on the volume wait queue until the operator retrieves the volume(s) and issues the *START SETUP command.

A customer's installation may not want to receive fetch messages for volumes in the library. To do so for common allocation requests (JCL statement allocation), install the SMC version of the JES3 user exit IATUX09. For dynamic allocation requests, install the SMC Type-1 modification to IATMDFE.

If you are running HSC, issue the ALLOC FETCH(OFF) or ALLOC FETCH(NONLIB) command. FETCH(OFF) is the default and suppresses fetch messages for any volume that is to be mounted on a library drive. If fetch messages are desired for nonlibrary volumes that are to be mounted on a library drive, FETCH(NONLIB) should be entered.

Note: FETCH(NONLIB) causes another volume lookup request to the HSC, which can affect performance.

If you are running MVS/CSC, issue the ALTER FETCH(NO) or ALTER FETCH(NONLIB) command. FETCH(NO) is the default and suppresses fetch messages for any volume that is to be mounted on a library drive. If fetch messages are desired for nonlibrary volumes that are to be mounted on a library drive, FETCH(NONLIB) should be entered.

Note: FETCH(NONLIB) causes another volume lookup request to the MVS/CSC, which can affect performance.

If your system is running with ALLOCATE=MANUAL as described above, when fetch messages are suppressed for a volume allocation, that allocation does not go onto the volume wait queue.

If your system is running with the SETPARAM statement set to FETCH=NO, or if you prefer to receive fetch messages for all volumes, then the IATMDFE Type-1 modification does not need to be applied to your system. The IATUX09 user exit also performs other functions and should be applied.

Drive Prioritization

The next step in Main Device Scheduler (MDS) allocates the devices required for the job. The SMC Type-1 modification to IATMDAL provides the SMC with the ability to review the list of drives available for each tape allocation. The list of drives contains drives that are online and available and are members of the group of drives defined in the esoteric placed in the Job Summary Table (JST) after drive exclusion processing.

The SMC drive priority is assigned based on the following criteria:

- For specific volumes, drives in LSMs closest to the volume are preferred.
- For scratch volumes, drives in LSMs with the largest number of scratch volumes are preferred.
- The TAPEREQ DEVTPREF parameter causes drives with specific recording techniques to be preferred.
- The TAPEREQ SCOPE parameter determines the relative priority of LSM location and drive type preferencing.

The following parameters, control statements, and user exits also influence the drive priority:

- HSC ALLOC command parameters LOWSCR, LSMPREF, and SCRTECH
- HSC SLSUX04 user exit.

After a final list of drives has been selected for allocation, the preference order of the eligible drives, after considering LSM and drive type preferencing, is selected based on a “last use” algorithm.

To reduce excessive wear on allocated drives, the SMC allocates drives by rotation based on the “last mount time” for each drive. This value is examined for every drive in the final drive list. The drive that had the most recent mount is located, and the drive immediately following it in the list is selected as the most preferred for the current allocation.

Note: This algorithm does not apply to virtual drives.

Deferring Mounts

After drive prioritization completes and all drives are allocated, the volumes required by the job, but not currently mounted, can be mounted before the job enters an initiator, during common allocation in the initiator, or at data set open time. Your installation may choose to defer the mount of library volumes.

The IATUX09 JES3 user exit modification must be installed to defer mounts to the initiator or to data set open time. This SMC modification interacts with the setting of the DEFER parameter of the HSC ALLOC command or the MVS/CSC ALTER command.

The following list describes the DEFER parameter settings and how the value determines what is deferred and when. The HSC value is listed first followed by the corresponding MVS/CSC value.

Defer(OFF/NO)
The volume mount is deferred only if the user's JCL requests it. This is the default.

Defer(JEs3)
The volume mount is deferred until the job enters the initiator. HSC SLSUX11 or MVS/CSC SCSUX11 can be used to override this value on an individual mount basis. For instance, the user exit can return a value to allow the mount to proceed before the job enters the initiator. JEs3 is the default value.

Defer(ON/YES)
The volume mount is deferred until the job enters the initiator and then again until the data set is opened. HSC SLSUX09 or MVS/CSC SCSUX09 can be used to override this value on an individual mount basis.

For optimal performance, StorageTek recommends setting Defer(ON/YES).

Note: Virtual mounts are always deferred.

JES3 Initialization Parameter Considerations

The library and nonlibrary drive environment must be defined to JES3 in the initialization deck using the following parameter statements:

- DEVICE statements to define drive addresses, device types, and XTYPEs
- SETNAME statements to define esoteric names and to associate them with XTYPEs
- HWSNAME statements to define the esoteric name relationships used during HWS processing.

This section describes these statements and shows how to code them for a sample configuration. Table 6 consists of the following drive addresses and esoterics attached to two systems, MVS1 and MVS2.

Table 6. Sample Configuration

Nonlibrary	ACS0	ACS1	Virtual
120-127 3480	220-223 4490	320-327 9490	A20-A5F Virtual
140-143 3490	240-243 9490	440-447 9490	
180-189 9840	280-289 9840	460-461 9940	

Note: The drive addresses and esoterics in this example are not meant to be taken literally but are intended to show how a wide variety of device types can be defined using JES3. Exact JES3 initialization statements are configuration dependent.

JES3 DEVICE Initialization Statements

DEVICE statements define the drives that JES3 can use to satisfy allocation requests. These statements define:

- drive addresses
- JES3/MVS systems that can access the drives
- initial drive online status
- device type of the drive.

The XTYPE parameter is especially important to SMC allocation. XTYPE connects devices with the same XTYPE value to a group of esoteric unit names. For example:

```
DEVICE,XTYPE=(DEV0220,CA),XUNIT=(220,MVS1,TAP,ON,220,MVS2,TAP,ON),
       NUMDEV=4,...
SETNAME,XTYPE=DEV0220,NAME=(CART,3490,SYS3480R,SYS348XR,ACS0TAP,...)
```

Devices 220-223 in ACS0 in the "Sample Configuration" on page 76 have been associated with the XTYPE name DEV0220. This name allows JES3 to allocate a device from the group 200-223 when any of the esoteric unit names listed on the SETNAME statement associated with XTYPE DEV0220 are specified in JCL or in a catalog entry.

The SMC relies on each XTYPE group to be unique with regard to real drive type and location. In the list of drives for ACS0, the 4490 drives should not be defined with the same XTYPE as the 9490 drives. Also, the 9840 drives located in ACS0 should not be defined with the same XTYPE as the nonlibrary 9840 drives.

During SMC initialization, XTYPE groupings are examined to verify these XTYPE restrictions. If an XTYPE contains mixed devices types or mixed locations, the characteristics of the first drive in the XTYPE group defines the remaining drives.

The SMC configuration report utility shows XTYPE, esoteric, and drive information. Refer to Chapter 8, "JES3 Configuration Report Utility" for more information about the configuration report.

The following example shows how DEVICE statements can be coded for this sample configuration.

Note: Drives must be defined to MVS prior to defining them to JES3. Use the Hardware Configuration Definition (HCD) facility to assign MVS unit addresses to the devices in the I/O Configuration.

```
* NONLIBRARY DRIVES
* 348Ø DRIVES 12Ø-127
DEVICE,XTYPE=(DEV012Ø,CA),JNAME=TNL,DTYPE=TA0348Ø,NUMDEV=8,
JUNIT=(Ø12Ø,MVS1,TAP,ON,Ø12Ø,MVS2,TAP,ON),
XUNIT=(Ø12Ø,MVS1,TAP,ON,Ø12Ø,MVS2,TAP,ON)
*
* 349Ø DRIVES 14Ø-143
DEVICE,XTYPE=(DEV014Ø,CA),JNAME=TNL,DTYPE=TA0349Ø,NUMDEV=4,
JUNIT=(Ø14Ø,MVS1,TAP,ON,Ø14Ø,MVS2,TAP,ON),
XUNIT=(Ø14Ø,MVS1,TAP,ON,Ø14Ø,MVS2,TAP,ON
*
* 984Ø DRIVES 18Ø-189 DEFINED AS 349Ø'S
DEVICE,XTYPE=(DEV018Ø,CA),JNAME=TNL,DTYPE=TA0349Ø,NUMDEV=1Ø,
JUNIT=(Ø18Ø,MVS1,TAP,ON,Ø18Ø,MVS2,TAP,ON),
XUNIT=(Ø18Ø,MVS1,TAP,ON,Ø18Ø,MVS2,TAP,ON)
*
* DRIVES IN ACSØ
* 449Ø DRIVES 22Ø-223
DEVICE,XTYPE=(DEV022Ø,CA),JNAME=ACSØ,DTYPE=TA0349Ø,NUMDEV=4,
JUNIT=(Ø22Ø,MVS1,TAP,ON,Ø22Ø,MVS2,TAP,ON),
XUNIT=(Ø22Ø,MVS1,TAP,ON,Ø22Ø,MVS2,TAP,ON)
*
* 949Ø DRIVES 24Ø-243
DEVICE,XTYPE=(DEV024Ø,CA),JNAME=ACSØ,DTYPE=TA0349Ø,NUMDEV=4,
JUNIT=(Ø24Ø,MVS1,TAP,ON,Ø24Ø,MVS2,TAP,ON),
XUNIT=(Ø24Ø,MVS1,TAP,ON,Ø24Ø,MVS2,TAP,ON)
*
* 984Ø DRIVES 28Ø-289 DEFINED AS 359Ø'S
DEVICE,XTYPE=(DEV028Ø,CA),JNAME=ACSØ,DTYPE=TA435901,NUMDEV=1Ø,
JUNIT=(Ø28Ø,MVS1,TAP,ON,Ø28Ø,MVS2,TAP,ON),
XUNIT=(Ø28Ø,MVS1,TAP,ON,Ø28Ø,MVS2,TAP,ON)
*
* DRIVES IN ACS1
* 949Ø DRIVES 32Ø-327
DEVICE,XTYPE=(ACS1949Ø,CA),JNAME=ACS1,DTYPE=TA0349Ø,NUMDEV=8,
JUNIT=(Ø32Ø,MVS1,TAP,ON,Ø32Ø,MVS2,TAP,ON),
XUNIT=(Ø32Ø,MVS1,TAP,ON,Ø32Ø,MVS2,TAP,ON)
*
*949Ø DRIVES 44Ø-447
DEVICE,XTYPE=(ACS1949Ø,CA),JNAME=ACS1,DTYPE=TA0349Ø,NUMDEV=8,
JUNIT=(Ø44Ø,MVS1,TAP,ON,Ø44Ø,MVS2,TAP,ON),
XUNIT=(Ø44Ø,MVS1,TAP,ON,Ø44Ø,MVS2,TAP,ON)
*
* 994Ø DRIVES DEFINED AS 359Ø'S
DEVICE,XTYPE=(DEV046Ø,CA),JNAME=ACS1,DTYPE=TA435901,NUMDEV=2,
JUNIT=(Ø46Ø,MVS1,TAP,ON,Ø46Ø,MVS2,TAP,ON),
XUNIT=(Ø46Ø,MVS1,TAP,ON,Ø46Ø,MVS2,TAP,ON)
*
* VIRTUAL DRIVES
DEVICE,XTYPE=(DEV0A2Ø,CA),JNAME=VIRT,DTYPE=TA0349Ø,NUMDEV=64,
JUNIT=(ØA2Ø,MVS1,TAP,ON,ØA2Ø,MVS2,TAP,ON),
XUNIT=(ØA2Ø,MVS2,TAP,ON,ØA2Ø,MVS2,TAP,ON)
```

JES3 SETNAME Initialization Statements

The SETNAME statements define all esoteric unit names and device type names associated with JES3-managed devices. These esoteric unit names and device type names can be specified by the `UNIT` parameter on a DD statement or as the unit type in a cataloged data set entry.

DEVICE statements associate a set of drives with an XTYPE. The SETNAME statement associates the XTYPE with a group of esoteric unit names.

During SMC esoteric unit name replacement, the relationships among the devices, the XTYPEs, and the esoteric unit names enable the SMC to choose the optimal esoteric unit name.

Note: For best SMC performance, each drive type in each location should have a unique esoteric defined. For example, an esoteric named A09840 could be defined to contain only the 9840 drives located in ACS0.

The following example shows how SETNAME statements can be coded for this configuration. The esoteric unit names specified in the NAMES parameter value list consist of the following:

- CART – all cartridge drives in the environment
- NLCART – all cartridge drives not in a library ACS
- A0CART – all cartridge drives in ACS0
- A1CART – all cartridge drives in ACS1
- ALL*xxxx* – all cartridge drives of the same device type, *xxxx*, independent of location
- LIB*xxxx* – all cartridge drives of the same device type, *xxxx*, in any library location
- *yyxxxx* – all cartridge drives of the same device type, *xxxx*, in location *yy*.

The generic device type names, such as 3480 or SYS3480R, are also specified in the NAMES lists.

```
* 3480/NONLIBRARY
SETNAME,XTYPE=DEV120,NAMES=(SYS3480R,CART,3480,NLCART,NL3480)
*
* 3490/NONLIBRARY
SETNAME,XTYPE=DEV0140,NAMES=(SYS3480R,SYS348XR,CART,3490,NLCART,
                             ALL3490, NL3490)
*
* 9840/NONLIBRARY
SETNAME,XTYPE=DEV0180,NAMES=(SYS3480R,SYS348XR,CART,3490,NLCART,
                             ALL9840,NL9840)
*
* 4490/ACS0
SETNAME,XTYPE=DEV0220,NAMES=(SYS3480R,SYS348XR,CART,3490,A0CART,
                             A04490,A0DEVT90)
*
* 9490/ACS0
SETNAME,XTYPE=DEV0240,NAMES=(SYS3480R,SYS348XR,CART,3490,A0CART,
                             ALL9490,LIB9490,A09490,A0DEVT90)
*
* 9840/ACS0
SETNAME,XTYPE=DEV0280,NAMES=(CART,3590-1,A0CART,ALL9840,A09840)
*
* 9490/ACS1
SETNAME,XTYPE=ACS19490,NAMES=(SYS3480R,SYS348XR,CART,3490,A1CART,
                              ALL9490,LIB9490,A19490)
*
* 9940/ACS1
SETNAME,XTYPE=DEV0460,NAMES=(CART,3590-1,A1CART,ALL9940,A19940)
*
* VIRTUAL DRIVES
SETNAME,XTYPE=DEV0A20,NAMES=(SYS3480R,SYS348XR,CART,3490,NLCART,
                             VIRTCART)
```

Refer to the appropriate version of the IBM *JES3 Initialization and Tuning Reference* for more information about esoteric unit name values for the SETNAME statement NAMES parameter.

JES3 HWSNAME Initialization Statements

The HWSNAME statements define which esoteric unit names are subsets of other esoteric unit names. Used during JES3 high watermark setup (HWS), these statements determine if a device can be reused from step to step.

The first HWSNAME TYPE parameter specifies the esoteric unit name, known as the *major name*, used during HWS processing. The following esoteric unit names, called *minor names*, can be used as an alternate to the major name.

The order of the minor names listed in the HWSNAME statement is the order in which they can be substituted for the major name. For example:

```
HWSNAME TYPE=(3490,ALL4490,ALL9490,ALL3490)
```

and

```
//STEP1 EXEC PGM...
//DD1   DD UNIT=3490,...
//STEP2 EXEC PGM...
//DD1   DD UNIT=ALL3490,...
//DD2   DD UNIT=ALL4490,...
```

JES3 HWS processing allocates two drives for this job. The Job Summary Table (JST) for the job after HWS shows the following esoterics for each DD allocation request:

- STEP1 DD1 and STEP2 DD2 JST entries contain ALL4490 because ALL4490 appears in the minor name list before ALL3490.
- STEP2 DD1 JST entry contains ALL3490.

Another example shows how HWS names are used when allocating across step boundaries:

```
//STEP1 EXEC PGM...
//DD1   DD UNIT=ALL9490,...
//DD2   DD UNIT=ALL4490,...
//STEP2 EXEC PGM...
//DD1   DD UNIT=3490
```

JES3 HWS begins with DD1 of STEP1 looking for an allocation in STEP2 that can use the same device. DD1 of STEP2 specifies 3490. The HWSNAME above for major name 3490 indicates that ALL9490 is an alternate (or minor) name for 3490. Therefore, STEP1 DD1 and STEP2 DD1 allocate the same drive. The JST entry for DD1 of STEP2 is not updated to reflect a new esoteric. The drive allocated for STEP1 DD2 is freed at the end of STEP1.

The minor names should not contain any devices that are not defined to the major name. For example:

```
HWSNAME TYPE=(A0CART,ALL9840,...)
```

Assume the following:

- A0CART contains drives 220-223, 240-243, and 280-289.
- ALL9840 contains drives 180-189 and 280-289.

ALL9840 contains drives (180-189) not in A0CART. In this case, volumes inside the library requesting a 9840 drive may attempt to allocate to a drive outside the library after HWS processing by JES3.

HWS processing occurs after SMC esoteric unit name replacement. Therefore, the HWSNAME definitions can affect the final allocation decision if JES3 also changes the esoteric unit name as in the first example.

The best solution for this situation is to create unique esoteric unit names (by location and device type) so that the SMC can select an esoteric unit name that has no minor name. See the HWSNAME entries in the following example that have been coded for the sample configuration.

```
* GENERIC MAJOR NAMES
HWSNAME TYPE=(SYS3480R)
HWSNAME TYPE=(SYS348XR)
HWSNAME TYPE=(3480,NL3480)
HWSNAME TYPE=(3490,SYS348XR,
              ALL3490,ALL9490,LIB9490,A0DEVT90,
              A04490,A09490,A19490,NL3490,NL9840)
HWSNAME TYPE=(3590-1, ALL9940,
              A09840,A19940)
*
* ALL DRIVES IN THE COMPLEX
HWSNAME TYPE=(CART,SYS3480R,SYS348XR,3490,3480,3590-1,
              ALL3490,ALL9840,ALL9490,ALL9940,LIB9490,
              A0CART,A1CART,NLCART,A0DEVT90,
              A04490,A09490,A09840,A19490,A19940,
              NL3480,NL3490,NL9840)
*
* DRIVES BY DEVICE TYPE
HWSNAME TYPE=(ALL3490,LIB9490,A0DEVT90,A09490,A19490,NL3490)
HWSNAME TYPE=(ALL9840,A09840,NL9840)
HWSNAME TYPE=(ALL9490,LIB9490,A09490,A19490)
HWSNAME TYPE=(ALL9940,A19940)
*
* DRIVES BY LOCATION
HWSNAME TYPE=(LIB9490,A09490,A19490)
HWSNAME TYPE=(NLCART,ALL3490,ALL3480,3480,
              NL3480,NL3490,NL9840)
HWSNAME TYPE=(A0CART,A04490,A09490,A09840,A0DEVT90)
HWSNAME TYPE=(A1CART,ALL9940,A19940,A19490)
*
* DRIVES BY LOCATION AND DEVICE TYPE
HWSNAME TYPE=(A0DEVT90,A04490,A09490)
HWSNAME TYPE=(NL3480)
HWSNAME TYPE=(NL3490)
HWSNAME TYPE=(NL9840)
HWSNAME TYPE=(A04490)
HWSNAME TYPE=(A09490)
HWSNAME TYPE=(A09840)
HWSNAME TYPE=(A19490)
HWSNAME TYPE=(A19940)
*
* VIRTUAL DRIVES
HSWNAME TYPE=(VIRTCART)
```

Device Preferencing Considerations

The DEVTpref parameter of the HSC and MVS/CSC TAPEREQ statement allows users to request a higher priority for one type of StorageTek 36-track drive during drive prioritization processing. A second or third model of 36-track drive can be specified as alternate choices. This device preferencing is applicable to a library configuration containing a mixture of 4490, 9490 and 9490EE cartridge drives.

To enable this processing, define an esoteric to include all the desired device types by ACS location or in the entire library configuration. In the sample configuration, the esoteric, A0DEVT90, serves this purpose for ACS0.

During drive exclusion, if a TAPEREQ indicated DEVT(9490,4490) for an allocation, the SMC could substitute A0DEVT90 for the original unit name if A0DEVT90 is a subset (e.g., UNIT=3490).

Note: JES3 HWS processing can change this esoteric to A09490 or A04490 when reusing drives across steps.

ZEROSCR Considerations

When specifying the HSC ALLOC command, the MVS/CSC ALTER command, or the MVS/CSC startup parameter ZEROSCR with a value of YES/ON, create esoteric unit names that span ACS boundaries. As an example, the following esoterics could be added to the sample installation:

- CA0A1 – an esoteric containing all drives in ACS0 and ACS1
- A0A1X490 – an esoteric containing all 4490 and 9490 drives in ACS0 and ACS1.

Assume both ACSs contain scratch volumes.

- If the scratch request does not specify media or recording technique, the SMC can substitute CA0A1 for CART.
- If the scratch request asked for 36-track recording technique, the SMC can substitute A0A1X490 for 3490.

In this way, both ACSs remain eligible for the allocation.

Note: Once again, JES3 HWS can alter esoteric unit names after the SMC has selected its choice.

Virtual Drive Definition Considerations

See the VTCS documentation for complete examples of JES3 initialization parameter statements for virtual drives.

SMC Normal Operations

The SMC runs on all processors that are active in a JES3 global and local environment. On both global and local processors, start the SMC and the library subsystem(s), the HSC, and/or MVS/CSC(s) before starting jobs requiring cartridge drives.

When the SMC and the library subsystem have initialized on the global processor and are communicating, the SMC performs drive exclusion, affinity separation, esoteric unit name replacement, fetch message suppression, drive prioritization, and mount deferral for both common and dynamic cartridge drive allocations. If the SMC has not completed initialization before jobs enter the JES3 C/I DSP, this processing is not performed. The PROMPT value on the NOSMC parameter of the SMCEHOOK macro delays one C/I DSP if the SMC has not initialized and reminds the operator to start the SMC.

When the SMC and the library subsystem have initialized on the local processor and are communicating, the SMC performs drive exclusion, affinity separation, and esoteric unit name replacement for dynamic cartridge drive allocations.

Notes:

- See the "Performing JES3 Post-Installation Tasks" appendix in the *NCS Installation Guide* for more information about the SMCEHOOK macro and its parameters.
- See Chapter 7, "Recovery Procedures" on page 137 for recovery procedures related to SMC, library subsystems, and JES3.

JES3 Constraints

Timing Between C/I and MDS

A time window exists between C/I processing and MDS processing. A requested volume's location or a scratch subpool count can change during the interval between these two processes. When this situation happens, one or more volumes may need to be ejected from or entered into an ACS.

JES3 High Watermark Setup and LSM Pass-Thru Processing

When a job consists of multiple steps, JES3 HWS processing attempts to minimize the number of devices required. Thus a job consisting of multiple steps, each requesting one tape drive, can be allocated a single drive for the entire job. The following example shows the possible effects on pass-thru processing.

Figure 4 shows a library configuration containing four LSMs. All drives in the library are online and available.

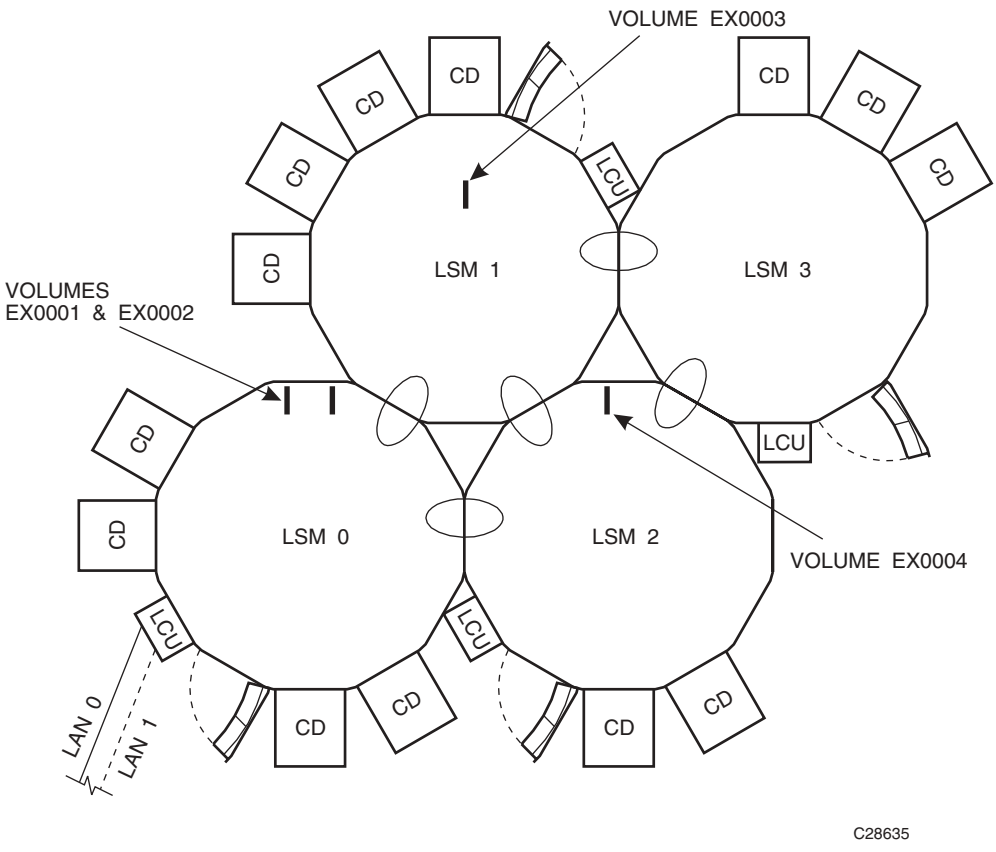


Figure 4. Volume Locations for the Pass-thru Example

The following example shows the JCL for the job:

```
//STEP1  EXEC
//DD1    DD DSN=DSN.IN.LSM0,UNIT=3490,VOL=SER=(EX0001,EX0002)
//*
//STEP2  EXEC
//DD1    DD DSN=DSN.IN.LSM1,UNIT=3490,VOL=SER=EX0003
//*
//STEP3  EXEC
//DD1    DD DSN=DSN.IN.LSM2,UNIT=3490,VOL=SER=EX0004
//*
//STEP4  EXEC
//DD1    DD DSN=DSN.IN.LSM0,UNIT=3490,VOL=SER=(EX0001,EX0002)
```

Volumes EX0001 and EX0002 are in LSM0, EX0003 is in LSM1, and EX0004 is in LSM2 and all volumes are the same media and require the same recording technique. The SMC drive exclusion process picked the same esoteric for the allocation.

After the SMC drive exclusion process completes, JES3 HWS analysis determines that the maximum number of drives required for running the job is one. MDS processing allocates the device. Pass-thru processing occurs as follows:

- If the allocated drive is attached to LSM0, the number of pass-thrus is two (volume EX0003 moves from LSM1, and volume EX0004 moves from LSM2).
- If the allocated drive is attached to LSM1 or LSM2, the number of pass-thrus is three (volumes EX0001 and EX0002 move from LSM0, and either EX0003 or EX0004 moves, depending upon which LSM contains the drive).
- If the allocated device is attached to LSM3, the number of pass-thrus is four (all volumes move to LSM3).

The SMC drive prioritization process uses the pass-thru counts when setting a priority for a drive. However, if the “preferred” drive is not available, other available drives can be selected.

Multiple Library Subsystem Support

When the SMC runs on an MVS host with multiple active library subsystems (i.e., HSC and/or MVS/CSCs) at the required NCS 5.1 or later release level, the SMC determines a library subsystem "owner" for each request.

The SMC builds and maintains a list of eligible library subsystems. The HSC subsystem, if active, always appears at the beginning of the list followed by all active MVS/CSC subsystems. Each subsystem is queried in list order to select the owner of the cartridge drive allocation. The following criteria, in the order listed, determine the "ownership" of the request.

1. Original esoteric or cataloged unit name
2. TAPEREQ esoteric
3. Subsystem User Exits 04 and 13
4. Specific volume location
5. Scratch volume availability.

If after evaluating all five criteria, no owner of the allocation request has been determined, the first subsystem in the available library subsystem list becomes the owner by default. The five criteria are explained below.

Original Esoteric or Cataloged Unit Name

The SMC examines the list of devices that belong to the unit name of the allocation request. The unit name can be located in a catalog entry for the data set specified in the allocation or can be explicitly defined using the UNIT parameter of the DD statement.

If no drives in the unit name are owned by a subsystem, that subsystem is excluded from consideration.

For example:

```
//DD1 DD DSN=THIS.DATASET,VOL=SER=CSC001,UNIT=HSCDRVS
```

Assume the following:

- Volume CSC001 resides in a library controlled by a MVS/CSC subsystem.
- All drives defined in esoteric HSCDRVS reside in a library controlled by the HSC subsystem.
- Both MVS/CSC and HSC are active.

The HSC owns the allocation request because it owns the drives.

TAPEREQ Esoteric

When the list of devices does not all belong to a single subsystem, a TAPEREQ esoteric can direct the allocation to a specific subsystem.

For example:

```
//DD1 DD DSN=CSC.DATASET,DISP=(NEW,KEEP),UNIT=3490
```

Assume the following:

- The unit name 3490 contains drives owned by MVS/CSC and owned by the HSC.
- The TAPEREQ below has been defined to the HSC.

  ```
  TAPEREQ DSN(CSC.DATASET) ESOTERIC(CSCDRVS) RECT(36B)
  ```

- The esoteric CSCDRVS contains all 3490 drives, and all are owned by the CSC.

The CSC owns the allocation request because it owns all the drives in esoteric CSCDRVS.

Subsystem User Exits 04 and 13

The SMC evaluates the HSC and MVS/CSC user exit policies to resolve allocation request ownership. If an MVS/CSC user exit requests a specific MVS/CSC, the requested MVS/CSC is always considered the request owner. If the user exit requests a nonlibrary allocation, or requests allocation to an esoteric containing no drives known to this subsystem, the SMC assumes that another subsystem may be the appropriate owner for the request. Other user exit return codes imply that this subsystem is the owner of the request.

See Appendix B, “JES3 User Exit Return Codes and Library Subsystem Ownership” for details of how user exit return codes determine the request owner.

Specific Volume Location

The SMC queries each active library subsystem to determine whether the volume resides in its library. The first subsystem that locates the volume owns the allocation request.

Scratch Volume Availability

The SMC queries each active library subsystem to determine the availability of the requested media, subpool, and label type. The first subsystem containing eligible scratch volumes owns the allocation request.

Note: The SMC performs identically to prior NCS allocation processing to determine availability of scratch volumes. Any subpool not defined to the library subsystem is treated as the general subpool (subpool 0). This subpool returns a “volume available” status if it contains scratch volumes.

SMC DFSMS Processing

The SMC interface to DFSMS provides the following capabilities:

- esoteric substitution during MVS JCL interpretation
- at device allocation time, the DFSMS DATACLAS value can request a specific recording technique and/or media.
- at device allocation time, the DFSMS MGMTCLAS value can be used for VTV allocations.
- at volume mount, the DFSMS MGMTCLAS value can be assigned to a VTV.
- at volume mount, the DFSMS DATACLAS value can request a media for a scratch request.

Enabling/Disabling the SMC DFSMS Interface

To enable the SMC DFSMS interface, specify the SMS parameter of the ALLOCDef command as follows:

```
ALLOCDEF SMS=ON
```

To disable the SMC DFSMS interface, specify the SMS parameter of the ALLOCDef command as follows:

```
ALLOCDEF SMS=OFF
```

The SMC DFSMS interface can be selectively enabled or disabled by job or job step by specifying the SMS parameter of the ALLOCJob command.

DATACLAS, STORCLAS, and MGMTCLAS Specification

DFSMS DATACLAS can be specified by using the DATACLAS parameter on the DD JCL statement or by executing an Automatic Class Selection (ACS) routine. STORCLAS and MGMTCLAS can only be specified by ACS routines.

Note: STORCLAS and MGMTCLAS JCL parameters are not supported by the SMC DFSMS interface due to conflicts with IBM MVS DFSMS. Using the STORCLAS JCL parameter causes a data set to become DFSMS-managed, and the MGMTCLAS JCL parameter requires a DFSMS-managed data set. The data sets assigned STORCLAS and MGMTCLAS values in the STKTAP1 environment are not actually DFSMS-managed.

Invoking ACS Routines

IBM DFSMS invokes ACS routines with the variable `&ACSENVIR` set to ALLOC before the SMC invokes the ACS routines with variable `&ACSENVIR` set to STKTAP1.

The SMC invokes the ACS routines at the following points in processing:

- SSI55 Interpreter/Dynamic Allocation Exit (IDAX)
- JES3 Converter/Interpreter (C/I)
- SSI23 JES3 Dynamic Allocation
- JES3 Main Device Scheduler (MDS)
- mount message interception.

The ACS routines are invoked in the following order:

1. data class
2. storage class
3. management class
4. storage group.

Management class and storage group ACS routines are called only if a storage class is assigned.

DFSMS Automatic Class Selection (ACS) Routine Environment for SMC

The following list of read-only variables is passed by SMC to DFSMS when the information is available to the SMC. Not all variables are available for every call to the ACS routines. In particular, processes that occur in the JES3 address space, such as MDS, do not provide the SMC access to the MVS control blocks that contain the values for these fields. See the descriptions of each DFSMS interface for exceptions.

- &ACSENVIR (equals STKTAP1 for the SMC interface)
- &ALLVOL
- &ANYVOL
- &DATACLAS
- &DD
- &DSORG
- &DSN
- &DSTYPE
- &EXPDT
- &FILENUM
- &JOB
- &LABEL
- &NVOL
- &PGM
- &RETPD
- &SYSNAME
- &SYSPLEX
- &UNIT.

In the STKTAP1 environment, the &ANYVOL variable is used only to match a specific VOLSER and does not contain the "REF=*xx*" values for VOL=REF allocations.

The &DATACLAS field is set when the JCL DD statement specifies this parameter.

Esoteric Substitution During IDAX

When a job is in the JCL interpretation phase, the unit name (esoteric) can be changed indiscriminately (for example, you could substitute disk for tape). Considerations:

- Esoteric substitution applies to all newly created data sets. New allocations are signified by specifying DISP=NEW, or in some cases, DISP=MOD in the JCL (refer to the IBM *MVS JCL Reference* manual for a list of requirements when using DISP=MOD to specify a new data set).
- Unit affinity chains are always separated if ACS routines return different esoterics for different members of the chain.
- VOL=REF chains within a job are validated and updated with the head-of-chain esoteric if necessary to ensure that volume references are honored.

When the SMC locates a new allocation that is not DFSMS-managed, the SMC calls the DFSMS ACS routines with the environment variable &ACSENVIR set to STKTAP1. When the ACS routines return a pair of storage group names with the first name being STK1, the second storage group name replaces the original value of the UNIT parameter of the DD statement. For example, the following DD statement allocates a 3490 nonlibrary tape drive (esoteric TNLIB):

```
//DD1  DD DSN=&TEMP.DSN,UNIT=TNLIB,DISP=NEW
```

Assume that your installation wants to move all new tape allocations into a tape library. You can code the DFSMS ACS routines to return storage groups STK1 and a library esoteric such as TACS0, where TACS0 contains all drives in library location ACS0. After esoteric replacement, the only drives considered for the allocation are contained within the TACS0 esoteric. The nonlibrary drives are no longer considered eligible for the allocation.

Defining Storage Groups and Storage Classes

To enable esoteric substitution during IDAX, complete the following steps:

1. Define a storage group of type POOL named STK1.
2. Define a storage group of type POOL for each possible esoteric returned, giving it the same name as the esoteric name.
3. Define at least one volume to each storage group. StorageTek recommends that you use a nonexistent volume for this definition.
4. Create a storage class ACS routine that, when it is invoked by the SMC (i.e., when the environmental variable is &ACSENVIR=STKTAP1), returns a storage class to be passed to the storage group ACS routine. Refer to Figure 5 on page 93.

5. Create a storage group ACS routine that, when the environmental variable is `&ACSENVIR=STKTAP1`, returns two storage groups (e.g., `&STORGRP='STK1','CART'`). In this case, "`STK1`" is the first storage group returned, and "`CART`," the replacement esoteric, is the second. Refer to Figure 5.
6. Test for the `&ACSENVIR=STKTAP1` read-only variable in the storage class and storage group ACS routines. This enables you to prevent the storage class being assigned when MVS invokes the ACS routines. If MVS invokes the SMC version of the ACS routines, the data set becomes DFSMS-managed and all SMC allocation functions are bypassed.
7. Return both a storage class and a storage group during the SMC invocation of the DFSMS ACS routines. If a storage class is returned but a storage group is not, DFSMS issues a message stating that allocation has failed, but in reality that may not be the case.

```
PROC STORCLAS

FILTLIST LOCALDSN INCLUDE (BACKUP*.**,
                           PROD.BKP*.**)

FILTLIST RMTDSN INCLUDE (PROD.OFFSITE.**)

  IF &ACSENVIR = 'STKTAP1' THEN
    SELECT
    WHEN &DSN = &LOCALDSN
      SET &STORCLAS = 'CART'
    WHEN &DSN = &RMTDSN
      SET &STORCLAS = 'RMTCART'
    END
  END

================================================================================

PROC STORGRP

  IF &ACSENVIR = 'STKTAP1' THEN
    SELECT
    WHEN &STORCLAS = 'CART'
      SET &STORGRP = 'STK1','CART'
    WHEN &STORCLAS = 'RMTCART'
      SET &STORGRP = 'STK1','RMTCART'
    END
  END
```

Figure 5. Creating Storage Class/Storage Group Routines

Availability of Read-only Variables

During DFSMS STORCLAS/STORGRP ACS routine processing, all read-only variables listed in "DFSMS Automatic Class Selection (ACS) Routine Environment for SMC" on page 91, except `&DSORG`, are available to the IDAX interface for esoteric substitution. The `&DATACLAS` field is set when the DD statement in the JCL specifies this parameter.

Validating DFSMS STORCLAS/STORGRP ACS Routine Execution

DFSMS provides the following ways to validate correct ACS routine execution:

- DFSMS WRITE statement in the ACS routines
- DFSMS ISMF test function.

When the SMC invokes the DFSMS ACS routines, all messages issued by DFSMS WRITE statements are routed to the GTF trace file if SMC tracing is active.

See the publication *DFSMS/MVS DFSMSdfp Storage Administration Reference* for further information about writing and testing DFSMS ACS routines.

Retrieving DFSMS Values During Allocation and Volume Mount

Requesting Recording Technique/Media using DFSMS DATACLAS

During allocation processing, the media and/or recording technique can be specified for a DD allocation by coding:

- TAPEREQ statement
- DATACLAS parameter on the DD statement
- DATACLAS ACS routines.

During scratch volume mount processing, only the volume media can be specified.

Note: DATACLAS media and recording technique values override TAPEREQ values.

Media/Rectech Support

The following media and recording techniques are supported:

Media:	
MEDIA1	Standard capacity for both 18-track and 36-track media types (this is the default)
MEDIA2	Enhanced capacity 36-track media
Recording Techniques:	
18TRACK	18-track recording technique
36TRACK	36-track recording technique

Defining Data Classes

To use DATACLAS, customers must define their own data class and specify the recording technique and media. If the recording technique is left blank, the SMC defaults to allowing both 18TRACK and 36TRACK to remain eligible for the allocation request, unlike MVS, which defaults to 36TRACK.

As an example, a customer can define a DATACLAS named STD18 that specifies a recording technique of 18TRACK and standard media.

When specifying a DATACLAS by coding it on the DD statement, the system programmer must have installed the optional Type-1 modification to IATIICM. The Type-1 modification retrieves the DATACLAS, determines its recording technique and media specification, and saves it in the IJSMEDIA field. This information is used during C/I processing (IATIIP1) and then passed to the JSTMEDIA field and used again during MDS processing.

When using an ACS routine to supply a DATACLAS to set recording technique and media, do not specify `&ACSENVIR='STKTAP1'`. The DATACLAS ACS routine is invoked both by the SMC and MVS, and the resulting recording technique and media should be consistent.

Refer to Figure 6 on page 95 for an example of how to define a data class routine.

```
PROC DATACLAS

FILTLIST RMTDSN INCLUDE(PROD.OFFSITE.**)

  SELECT
  WHEN &DSN = &RMTDSN
    SET &DATACLAS = 'ECART36'
  END
```

Figure 6. Creating Data Class Routines

In the example above, assume DATACLAS ECART36 is defined with a recording technique of 36-track and a media type of MEDIA2. This data class routine example assigns ECART36 to data sets with names that start with "PROD.OFFSITE."

By default, any DATACLAS value specified by the DFSMS ACS routine overrides the DATACLAS parameter specified on the JCL statement. Your installation can change this behavior by adding the following statements to your DFSMS DATACLAS ACS routine.

```
WHEN (&DATACLAS NE '')
DO
  SET &DATACLAS = &DATACLAS
EXIT
END
```

Note: During JES3 C/I POSTSCAN processing, JES3 MDS processing, and IAT5210 mount message processing, the DATACLAS name specified on the JCL statement is not available and is not passed to the ACS routines. Thus, the statements above may not produce the expected results.

Requesting Management Class Using DFSMS ACS Routines

During allocation and mount processing, a management class name can be specified for a virtual allocation by coding any of the following:

- TAPEREQ statement
- MGMTCLAS ACS routine.

The MGMTCLAS JCL parameter is not supported by the SMC DFSMS interface.

Note: The MGMTCLAS management class name overrides a TAPEREQ management class name.

When writing a management class routine, keep in mind:

- The management class routine is invoked only when a storage class is assigned.
- The management class routine must test for the `&ACSENVIR='STKTAP1'` read-only variable value.

DFSMS issued messages for JES3 static allocations are routed to the SMC GTF trace file.

Refer to Figure 7 on page 97 for an example of how to define a management class routine.

```
PROC STORCLAS

FILTLIST CART INCLUDE ('CART')

  IF &ACSENVIR = 'STKTAP1' THEN
    SELECT
    WHEN (&UNIT EQ &CART)
      SET &STORCLAS = 'VIRTAPE'
    WHEN (&UNIT NE &CART)
      SET &STORCLAS = 'STKDFLT'
    END
  END

/*====================================================================*/

PROC STORGRP

  IF &ACSENVIR = 'STKTAP1' THEN
    SELECT
    WHEN (&STORCLAS = 'VIRTAPE')
      SET &STORGRP = 'STK1', 'VDRIVES'
    END
  END

/*====================================================================*/

PROC MGMTCLAS

FILTLIST LOCAL INCLUDE(BACKUP*.**.,
                       PROD.BKP*.**)

FILTLIST REMOTE INCLUDE(PROD.OFFSITE.**)

  IF &ACSENVIR = 'STKTAP1' THEN
    SELECT
    WHEN (&DSN = &LOCAL)
      SET &MGMTCLAS = 'INVTAPE'
    WHEN (&DSN = &REMOTE)
      SET &MGMTCLAS = 'OFFVTAPE'
    END
  END
```

Figure 7. Creating Management Class Routines

Note: In this example, the STORCLAS routine assigns a storage class to every call. This ensures that the MGMTCLAS routine is also driven at mount time. Only the VTCS component of NCS currently uses the management class value.

Availability of Read-only Variables

During DFSMS ACS routine processing, the SMC sets the values for all read-only variables when the information is available. Not all information is available for each process for which the SMC calls ACS routines.

During JES3 C/I POSTSCAN processing, the following read-only variables are **not** available to the SMC:

- &DATACLAS (if specified in JCL)
- &EXPDT
- &PGM
- &RETPD.

During SSI23 dynamic allocation processing, all read-only variables are available to the ACS routines.

During JES3 MDS processing, the following read-only variables are **not** available to the SMC:

- &DATACLAS (if specified in JCL)
- &EXPDT
- &PGM
- &RETPD.

During IAT5210 mount message processing, the following read-only variables are **not** available to the SMC:

- &DATACLAS (if specified in JCL)
- &EXPDT
- &PGM
- &RETPD
- &UNIT (does not contain the original esoteric but does contain the selected device number, such as 0A10).

When processing mount message IEF233A for a dynamic allocation, the `&DATACLAS` value specified by the SVC99 text unit DADACL is not available.

When processing mount message IEC501A, the SMC sets the `&UNIT` read-only variable to the generic unit type, such as 3490.

Note: Each SMC interaction with DFSMS invokes all levels of ACS routines. The variable availability listed above applies to all ACS routines.

Validating DFSMS ACS Routine Execution

DFSMS provides the following ways to validate correct ACS routine execution:

- DFSMS WRITE statement in the ACS routines
- DFSMS ISMF test function.

When the SMC invokes the DFSMS ACS routines, all messages issued by DFSMS WRITE statements or by DFSMS processing are routed to the GTF trace file if SMC tracing is active. During SSI23 Dynamic Allocation processing and MVS mount message processing, these messages are also issued to the SYSMSG data set of the job.

See the publication *DFSMS/MVS DFSMSdfp Storage Administration Reference* for further information about writing and testing DFSMS ACS routines.

Interaction with Other Software

Open Type J

The SMC allocation enhancements may not operate if you use the MVS Open Type J macro. Since that macro allows you to change volume serial numbers or data set names at open time, information available at Job Step Allocation time may be incorrect as interpreted by the SMC.

Note: Some vendor software products use MVS Open Type J. If you are experiencing unexpected allocation results using a vendor software product, check with the vendor to determine if Open Type J is used and follow the recommendations below.

SMC allocation may be influencing MVS allocation erroneously based on information that may have changed at open time. To prevent this problem when using the Open Type J macro, specify the appropriate esoteric in the JCL or specify an appropriate esoteric in an applicable TAPEREQ control statement.

SAMS: DISK (DMS)

Sterling Software's SAMS: DISK (DMS) has two methods of allocating transports:

- allocates transports at session start-up, holds onto the transports throughout the session and uses Open Type J (refer to the section about ''Open Type J'').
- uses dynamic allocation (DYNALLOC) to allocate transports when required.

The SMC allocates correctly when dynamic allocation is used. Therefore, the latter method of allocating transports is recommended.

Differences Between SMC and NCS Allocation

Drive Exclusion

The SMC influences allocation incrementally based on an ordered list of exclusion levels. The SMC allows a job to fail when not enough drives remain eligible after performing the minimum exclusion level. The HSC and MVS/CSC do not allow a job to fail before initiation, but the job may fail at execution time due to incompatible drive selection.

Recording Technique Exclusion

The SMC uses the requested recording technique as the last criterion in drive exclusion (exclusion level 8). If your installation has been using TAPEREQ or SMS recording technique to control allocation, such as specifying RECTECH(18TRACK) to direct allocation outside the library, you may need to change your TAPEREQ parameters. For example, if the library contains available scratches, all nonlibrary drives are excluded at exclusion level 6, and the allocation goes inside the library.

Note: StorageTek recommends that TAPEREQ statements specifying an 18-track or 36-track recording technique be changed to use an esoteric instead. Since the SMC allows any valid esoteric substitution, specifying TAPEREQ ESOTERIC(3480), which is applied at exclusion level 4, accomplishes the goal of directing the allocation to a 3480 device.

Bypassing Jobs

The SMC can bypass allocation influence for individual jobs if requested. The SMC ALLOCJob command BYPASS parameter allows a job to run as if no SMC and supporting HSC or MVS/CSC subsystem(s) are present.

Unit Affinity

The HSC ALLOC command and MVS/CSC ALTER command parameters GDGALL and UNITAFF are no longer supported and have been replaced by the SMC ALLOCDef and ALLOCJob command SEPLVL parameter during drive exclusion processing.

SEPLVL applies to both unit affinity and GDGALL chains. The default value for SEPLVL, MAX, always separates chains if no common eligible drives remain after drive exclusion has processed each member of the chain for all exclusion levels.

If your installation previously operated with GDGALL and UNITAFF set to NOSEP and would like the SMC to operate similarly, specify:

```
ALLOCDEF SEPLVL(3)
```

If your installation previously operated with UNITAFF and GDGALL set to SEP, then the default SEPLVL value (MAX) provides comparable functionality.

Note: The HSC and MVS/CSC UNITAFF parameter value still provides unit affinity separation decisions when using DFSMS storage group name processing to replace the unit name during JCL interpretation.

User Exit Parameters

The SMC does not honor the following HSC ALLOC or MVS/CSC ALTER command parameters:

- UXPRMS (HSC)
- XJ3SUB (MVS/CSC).

The SMC operates the same as the HSC and MVS/CSC did when Honor was specified for the UXPrms parameter of the HSC ALLOC command or the XJ3sub MVS/CSC startup parameter.

User Exit Processing

In NCS allocation, User Exit 12 (Unit Affinity Separation Exit) is called for each chain. If the results of the call create two or more new chains, User Exit 12 is also called for the new chains. The exit is called until no further chain separation occurs.

In SMC allocation, User Exit 12 is called only once for each original chain. User Exit 13 (Specific Volume Exit) is called for each specific volume in the chain and User Exit 04 (Scratch Volume Exit) is called for each scratch request in the chain. Refer to “Affinity Separation” on page 66 for more information about unit affinity chain processing.

Esoteric Unit Name Substitution

The SMC uses the JES3 internal tables of XTYPEs and esoterics to select the best possible esoteric for each DD statement. The SMC does not require a TAPEREQ or user exit-supplied esoteric to be a subset of the original unit name, nor does it require the esoteric name to be a defined as major name on a HWSNAME JES3 initialization statement. The intersection of drives between the original unit name and the customer-supplied esoteric is used to choose the replacement esoteric. Refer to “Esoteric Unit Name Replacement” on page 70 for more information about the esoteric replacement process.

DFSMS IDAX Esoteric Substitution

The SMC DFSMS IDAX interface call to the ACS routines (`&ACSENVIR=STKTAP1`) follows the IBM call to the ACS routines (`&ACSENVIR=ALLOC`).

Unit affinity chains are always separated when members of the chain receive different esoteric values.

Volume reference chains always propagate the esoteric received for the first use of the volume throughout the chain.

DFSMS DATACLAS recording technique and media support is no longer restricted to installations that only have a StorageTek tape library. If your installation has a mixture of VTS and StorageTek tape libraries, you can enable the StorageTek DFSMS DATACLAS support by installing the IATIICM type-1 modification.

Messages

The MSGDEF or MSGJOB command can suppress SMC messages by severity level.

HSC message SLS1350I and MVS/CSC message SCS1350I are replaced with SMC messages SMC0045 and SMC0046. These messages display conflicting allocation criteria.

SMC0073 replaces HSC message SLS0954D issued when a job enters C/I, and the JES3 allocation intercept is not yet active. Message SMC0073 is issued if the SMCEHOOK macro specified `NOSMC=PROMPT` when the macro was compiled into IATIIP1.

Virtual Esoteric Unit Names

See the VTCS documentation for information regarding requirements when defining virtual esoteric unit names.

Chapter 5. SMC Message Handling

Overview

The SMC intercepts specific MVS, JES3, and Tape Management System (TMS) messages related to mount, dismount, and swap operations. When the intercepted message includes a drive defined to a library subsystem (HSC or MVS/CSC), the SMC directs the owning library subsystem to perform the requested operation.

Messages that are intercepted by the SMC are listed in Appendix C, "Intercepted Messages" on page 189.

Note: The IATUX71 user exit must be installed to allow JES3 mount messages to be processed. See the *NCS Installation Guide* for more information.

User Directed Message Handling

If the SMC does not currently support the Tape Management System at your installation, you can still direct the SMC to intercept the specific messages issued by your TMS. Use the USERMsg operator command to define these additional messages. See "USERMsg Command" on page 135 for more information.

The HSC and MVS/CSC each provide a user exit (SLSUX01 and SCSUX01, respectively) to allow examination of a message and to direct the type of processing (mount, dismount, or swap) for the message. See the *HSC System Programmer's Guide* and *MVS/CSC System Programmer's Guide* for more information about these user exits.

The SMC calls the library subsystem user exit for each intercepted message. This includes the default messages listed in Appendix C, "Intercepted Messages"and all messages defined using the USERMsg command.

Notes:

- Unlike previous NCS releases, only messages intercepted by the SMC are passed to the user exit.
- The SMC does not support the user exit return code of REPLY.

Message Handling Policies

The SMC honors the following MVS, HSC and MVS/CSC policies related to mount, dismount, and swap message handling:

MVS Policies

The System Authorization Facility (SAF) can be used to protect tapes at the volume level (CLASS=TAPEVOL) using your current security software. If defined, the SMC honors the policies defined through the SAF interface regarding the write-protect requirement for a volume mounted on a library transport. The SMC invokes the SAF interface by issuing a RACROUTE macro, and protects read-only volumes through the ACS Virtual Thumbwheel (VTW) support.

HSC Policies

The LIBGEN delete disposition policy is honored by the SMC to control whether volumes dismounted by MVS with the delete disposition are to be scratched.

The MNTD MOuntmsg(Roll|Noroll) policy is honored by the SMC to determine whether mount messages handled by the SMC are allowed to roll off the operator console screen before the mount request is satisfied.

The MNTD VOLWatch(Off|ON) policy is honored by the SMC to determine whether the SMC issues an SMC0106 message whenever a library volume is requested to be mounted on a non-library device.

The AMPND startup parameter is honored by the SMC to determine whether pending mounts for HSC owned drives are automated when the SMC is initialized.

MVS/CSC Policies

The ALTER DELDisp(SCRTCH|NOSCRTCH) policy is honored by SMC to control whether volumes dismounted by MVS with the delete disposition are to be scratched.

The ALTER WTOdesc(YES|NO) policy is honored by the SMC to determine whether mount messages handled by the SMC are allowed to roll off the operator console screen before the mount request is satisfied.

The AMPND startup parameter is honored by SMC to determine whether pending mounts for MVS/CSC owned drives are automated when the SMC is initialized.

Tape Management System Support

The SMC interprets Mount, Dismount, and Swap messages from the following Tape Management Systems:

- CA-1
- CA-DYNAM/TLMS
- DFSMSrmm
- AutoMedia (Zara)
- CONTROL-T

For tape management systems that supply a subpool, the subpool is interpreted by the SMC and used as the requested subpool name, unless overridden by user exit 01 or a TAPEREQ statement. Related messages include:

- CTS002
- CTT101A
- CTT104A
- TMS002

SMC Swap Processing

The SMC automates the I/O error initiated swap process in the same manner as HSC enhanced swap processing.

The SMC swap process begins when one of the following messages is issued:

```
IGF500I SWAP  XXX1 TO XXX2 - I/O ERROR

IGF503I ERROR ON XXX1, SELECT NEW DEVICE

IGF509I SWAP XXX1 - I/O ERROR
```

If device *XXX1* is known to the SMC as a library or nonlibrary defined device, the SMC suppresses the message and begins the automatic swap process.

The SMC issues one of two messages:

```
SMC0108  No compatible drive found for SWAP processing
```

Or when a compatible drive can be selected by the SMC:

```
SMC0107  SWAP volser from XXX1 to XXX2
```

Device *XXX2* is the SMC-selected device that has been determined to be compatible for the swap. The SMC next suppresses the MVS IGF500D or IGF509D message and replaces the message with:

```
SMC0110  Allow swap of volser from XXX1 to XXX2;
Reply 'Y', or 'N' or DEVICE
```

The operator may approve the device selected, cancel the swap or choose a different device. If the operator selects a different device, the SMC accepts the device with no further compatibility checking.

If the reply is 'Y' or a new device, MVS issues the following message:

```
IGF502E  PROCEED WITH SWAP OF XXX1 TO XXX2
```

If *XXX1* is a library owned device, the dismount of the volume is automated.
If *XXX2* is a library owned device, the mount of the volume is automated.

HSC Mount-Related Messages

Certain mount-related messages may still be issued by the HSC due to error conditions.

- SLS0088D is issued when, due to an error condition, a repeated mount for the same volume is required.
- SLS1075D is issued when a dismounted volume encounters an I/O or other type of error.

In addition, the following HSC messages are replaced by SMC messages:

- SLS4306I and SLS4308I are replaced by SMC0106.
- SLS4310D is replaced by SMC0110.

Chapter 6. Operator Commands

Overview

This chapter provides information for the following SMC commands:

- ALLOCDef
- ALLOCJob
- CMDDef
- LIst
- MSGDef
- MSGJob
- READ
- RESYNChronize
- TRace
- USERMsg

Issuing SMC Commands

SMC operator commands can be issued using either the MVS Modify command, or a user-defined command prefix defined in the SMC CMDDef command.

- Issuing SMC commands using the MVS Modify command
- Issuing SMC commands using the SMC command prefix
- Specifying SMC commands in the SMCPARMS or SMCCMDS data set.

Issuing SMC Commands Using the MVS Modify Command

The following information is included when an SMC command is issued using the MVS Modify (F) command:

- MVS Modify command (F)
- SMC started task name
- command name
- parameters (optional or required).

SMC commands are entered in the following format:

```
F started-task-name,command-name [parameter]
```

The following example illustrates the MSGDef command:

```
F SMC1MVS,MSGDEF LVL=4
```

Notes:

- The started task name and command name **must** be separated with a comma. Spaces are **not** allowed between the subsystem name and command name.
- Parameters and values can be separated with any number of spaces, and may optionally include an equal (=) sign. Values may also be included in parentheses.

Issuing SMC Commands Using the SMC Command Prefix

SMC operator commands can be issued using an SMC command prefix. This prefix is defined using the SMC CMDDef command. See "CMDDef Command" on page 119 for more information about this command.

Specifying SMC Commands in the SMCPARMS or SMCCMDS Data Set

SMC operator commands can be specified in the SMCPARMS or SMCCMDS data set. These commands are automatically processed at startup. See "Control Statements" on page xx for syntax conventions used when specifying commands in the SMCPARMS or SMCCMDS data set.

- The SMCCMDS data set can be reprocessed while the SMC is active using the READ operator command. The following is a sample SMCCMDS member entry:

```
MSGDEF CASE(MIXED) LVL(4)
ALLOCDEF MINLVL=4 SEPLVL=MIN
```

- The SMCPARMS data set is used for user-configured items that cannot be changed while the SMC is active. The SMCPARMS data set **cannot** be reprocessed using the READ command. The following is a sample SMCPARMS member entry:

```
CMDDEF PREFIX(B@F$)
```

Note: StorageTek recommends that you include **only** the CMDDef PREFix and USERMsg ID parameter settings in the SMCPARMS data set. Specify all other commands in the SMCCMDS data set.

Operator commands that allow specification of JOBname, STEPname, or PROCstep are evaluated in order of most specific to least specific job name specification. Therefore, commands can be entered in any order. Consider the following example:

Two ALLOCJob commands are entered:

```
ALLOCJOB JOBNAME=NOALLOC* MINLVL=4
ALLOCJOB JOBNAME=NOALLOC1 MINLVL=3
```

Regardless of the order in which these commands are entered, job name NOALLOC1 is processed with MINLVL 3 because this command's job name is more specific than job name NOALLOC*.

SMC Command Descriptions

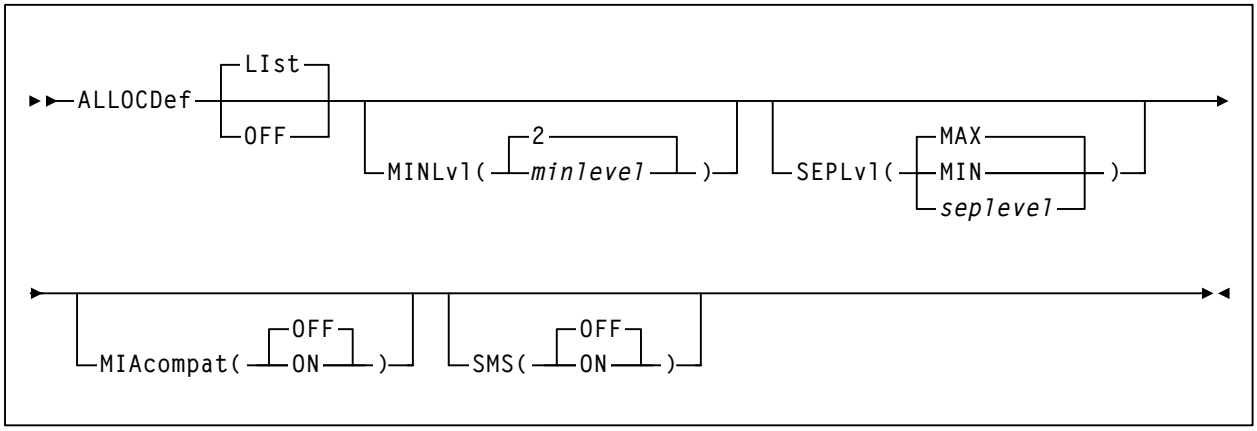
This section describes the SMC operator commands. Command syntax, parameter descriptions, and examples are included for each command.

See “Syntax Flow Diagrams” on page xv for syntax flow diagramming conventions.

ALLOCDef Command

The SMC ALLOCDef command is used to specify system default parameters used within the SMC subsystem to control allocation.

Syntax



Command Name

ALLOCDef
initiates the ALLOCDef command.

Parameter Descriptions

LIst
optionally, lists current default allocation settings.

This is the default if no parameters are specified. It **cannot** be specified with any other parameter.

OFF
optionally, resets all default allocation values to original SMC default settings. It **cannot** be specified with any other parameter.

MINLvl

optionally, specifies the desired minimum level of drive exclusion. If a job is not allocatable at the minimum exclusion level, the SMC still excludes drives to the minimum level and allows the job to fail.

minlevel

indicates the desired minimum allocation exclusion level. Valid values are 1-8. See Chapter 3, "SMC Allocation in a JES2 Environment" or Chapter 4, "SMC Allocation in a JES3 Environment" for more information regarding SMC exclusion levels.

Note: Increasing ***minlevel*** has no effect unless message SMC0045 or SMC0091 has been issued for a specific job step.

SEPLvl

optionally, specifies the exclusion level at which affinity and GDG chains are separated.

Note: If HSC or MVS/CSC allocation previously specified UNITAFF(NOSEP) and GDGALL(NOSEP), the default SMC exclusion tables can specify SEPLvl=3 in order to preserve existing behavior for scratch affinity chains with different media types.

MIN

do **not** separate chains beyond the minimum level.

MAX

separate chains for conflicting exclusion criteria whenever sufficient drives are available. This is the default value.

seplevel

indicates a separation level between the minimum and maximum levels. When sufficient drives exist, the SMC attempts to separate chains at this level. See Chapter 3, "SMC Allocation in a JES2 Environment" or Chapter 4, "SMC Allocation in a JES3 Environment" for more information regarding SMC exclusion levels.

Note: The *seplevel* value **cannot** be less than the *minlevel* value.

MIAcompat

optionally, specifies whether or not the EDL is to be modified at SSI24 time for compatibility with Computer Associates Unicenter CA-MIA Tape Sharing for z/OS and OS/390 product. This parameter is not valid in a JES3 with SETUP environment.

OFF

do not update the EDL at SSI24 time. This is the default value.

ON
update the EDL at SSI24 time. Specify this value if you use Unicenter CA-MIA.

SMS
enables and disables the DFSMS interface. When the DFSMS is enabled, SMC invokes the DFSMS ACS routines. The returned constructs are used to influence device allocation of data sets.

OFF
disables the DFSMS interface. This is the default value.

ON
enables the DFSMS interface.

Example

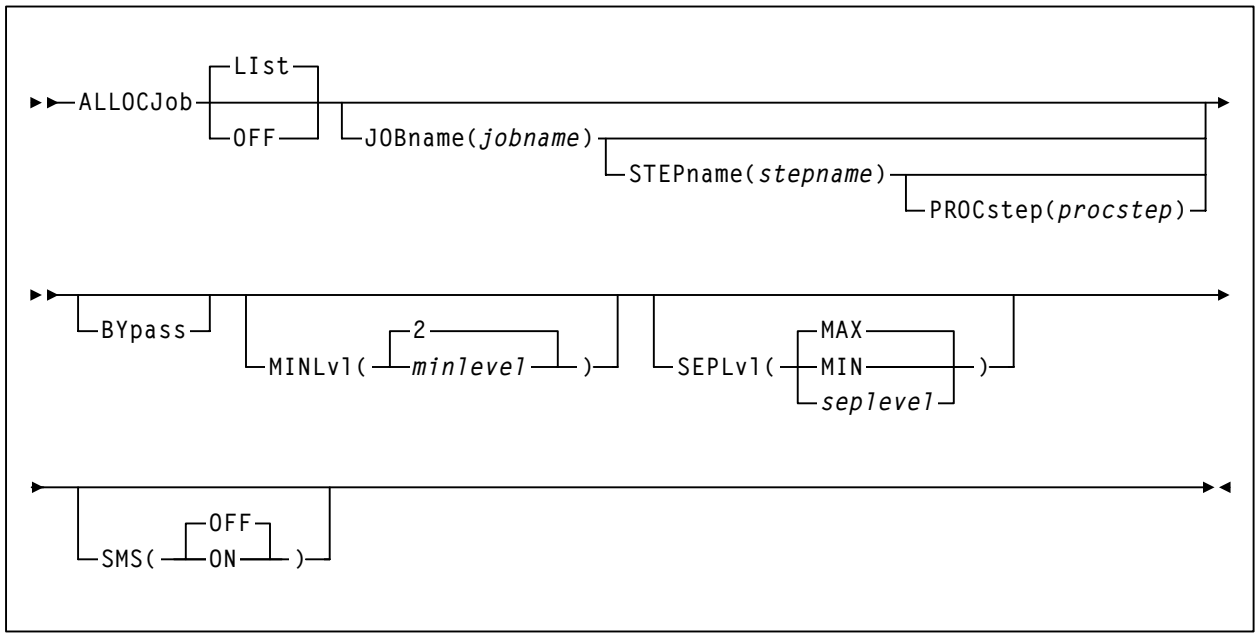
In the following example, the ALLOCDef command sets the minimum level of allocation exclusion to 4 and specifies that unit affinity and GDGALL chains are separated based on levels 1-4.

```
ALLOCDEF MINLVL=4 SEPLVL=MIN
```

ALLOCJob Command

The SMC ALLOCJob command is used to override SMC default allocation parameters by job name, step name, and PROC step.

Syntax



Command Name

ALLOCJob
initiates the ALLOCJob command.

Parameter Descriptions

LIst
optionally, lists current default allocation settings and override settings (in the order they are processed) by job name, step name, and PROC step.

This is the default if no parameters are specified. It **cannot** be specified with any other parameter.

OFF
optionally, removes all job name, step name, and PROC step overrides for allocation.

- If no other parameters are specified, allocation overrides are removed for all jobs.
- If only JOBname is specified, allocation overrides are removed for ALL ALLOCJob entries for that job name.

- If JOBname, STEPname and PROCstep are specified, allocation overrides are removed **only** for the specified entry.

Global allocation settings are not affected by this parameter.

JOBname
optionally, specifies a job name.

This parameter is required unless LIst or OFF is specified.

jobname
indicates the job name.

The value entered **must** be one to eight characters in length. An asterisk (*) can be used as a wildcard when included as the last character. ALLOCJob is set for all jobs whose job names match the characters preceding the asterisk.

STEPname
optionally, specifies a step name. This parameter is **only** valid when JOBname is also specified. It is **not** valid in JES3 environments with TAPE SETUP processing.

stepname
indicates the step name.

The value entered **must** be one to eight characters in length. An asterisk (*) can be used as a wildcard when included as the last character.

PROCstep
optionally, specifies a PROC step. This parameter is **only** valid when JOBname and STEPname are also specified. It is **not** valid in JES3 environments with TAPE SETUP processing.

procstep
indicates the PROC step.

The value entered **must** be one to eight characters in length. An asterisk (*) can be used as a wildcard when included as the last character.

BYpass
optionally, specifies that SMC allocation influencing is **not** performed for the indicated job (job step, step name, PROC step).

MINLvl
optionally, specifies the desired minimum level of drive exclusion. If a job is not allocatable at the minimum exclusion level, the SMC still excludes drives to the minimum level and allows the job to fail.

This parameter **cannot** be specified with the BYpass parameter. It is **only** valid when JOBname is also specified.

minlevel

indicates the desired minimum drive exclusion level. Valid values are 1-8. See Chapter 3, “SMC Allocation in a JES2 Environment” or Chapter 4, “SMC Allocation in a JES3 Environment” for information regarding SMC exclusion levels.

Note: Increasing ***minlevel*** has no effect unless message SMC0045 or SMC0091 has been issued for a specific job step.

SEPLvl

optionally, specifies the exclusion level at which affinity and GDG chains are separated.

This parameter **cannot** be specified with the BYpass parameter. It is **only** valid when JOBname is also specified.

MIN

do **not** separate chains beyond the minimum level.

MAX

separate chains for conflicting exclusion criteria whenever sufficient drives are available. This is the default value.

seplevel

indicates a level between the minimum and maximum levels. When sufficient drives exist, the SMC attempts to separate chains at this level. See Chapter 3, “SMC Allocation in a JES2 Environment” or Chapter 4, “SMC Allocation in a JES3 Environment” for more information regarding SMC exclusion levels.

Note: The *seplevel* value **cannot** be less than the *minlevel* value.

SMS

enables and disables the DFSMS interface. When the DFSMS is enabled, SMC invokes the DFSMS ACS routines. The returned constructs are used to influence device allocation of data sets.

OFF

disables the DFSMS interface. This is the default value.

ON

enables the DFSMS interface.

Example

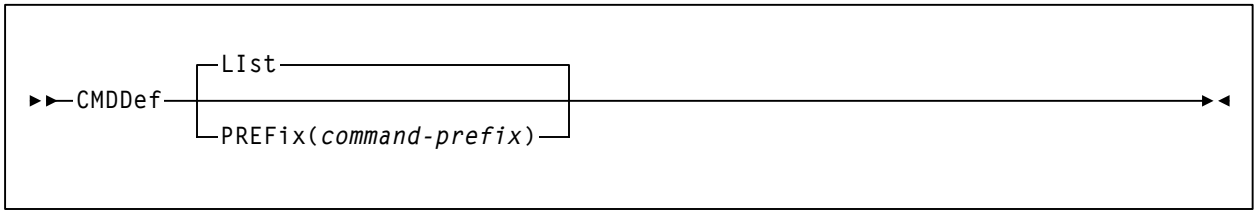
In the following example, the ALLOCJob command specifies that jobs with stepnames beginning with “STEP1” are bypassed during allocation influencing. These are the **only** steps that are bypassed.

```
ALLOCJOB JOB=PRODJOB1 STEP=STEP1* BYPASS
```

CMDDef Command

The SMC CMDDef command is used to assign an SMC command prefix.

Syntax



Command Name

CMDDef
initiates the CMDDef command.

Parameter Descriptions

LIst
optionally, lists the current command prefix.

This is the default if no parameters are specified. It **cannot** be specified with any other parameter.

PREFix
optionally, specifies a command prefix for the SMC subsystem.

Note: This parameter may **only** be specified in the SMCPARMS data set.

command-prefix
indicates the desired command prefix characters.

The value entered **must** be one to eight characters in length and **must** meet the following requirements:

- Valid characters include:

  A-Z 0-9 @ ¢ $ # , . / ‘ ( ) < > * & + - = | ! ; : “ % _ ?

- The command prefix **cannot** include a command string, a command abbreviation, or any string that invokes a command.
- The command prefix **cannot** include a string that is a subset or superset of an existing prefix beginning with the same character.

Note: The MVS command, DISPLAY OPDATA, displays all active command prefixes and their corresponding subsystem name.

Example

In the following example, the CMDDef command is specified in the SMCPARMS data set as follows:

```
CMDDEF PREFIX(B@F$)
```

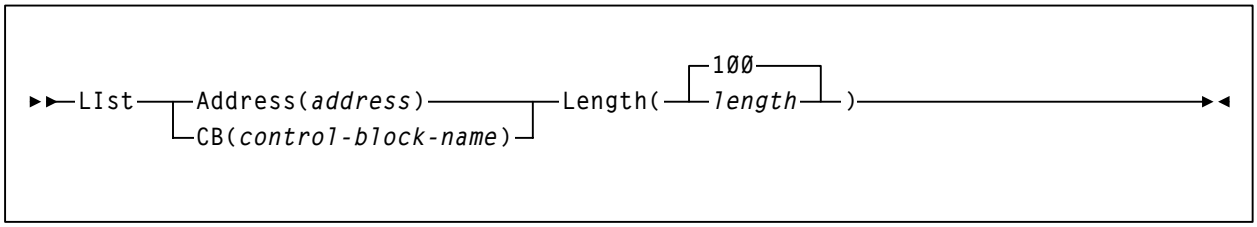
As a result, SMC operator commands can be entered as follows:

```
B@F$MSGDEF CASE(MIXED)
```

LIst Command

The SMC LIst command is used to display storage accessible from the SMC started task address space.

Syntax



Command Name

LIst
initiates the LIst command.

Parameter Descriptions

Address
specifies the address at which to begin listing SMC memory contents.

address
indicates the address. The value entered **must** be a valid hexadecimal address.

CB
specifies the internal SMC control block to be listed.

control-block-name
indicates the control block name.

SMC control blocks are listed for diagnostic purposes. Specify *control-block-name* only as directed by StorageTek Software Support.

Length
optionally, specifies the length of SMC memory (bytes) to be listed.

length
indicates the length. The value entered **must** be a valid hexadecimal value between 1 and FFFF. Memory is listed beginning at the location specified in the *address* parameter. The default value is 100 (decimal 256).

Example

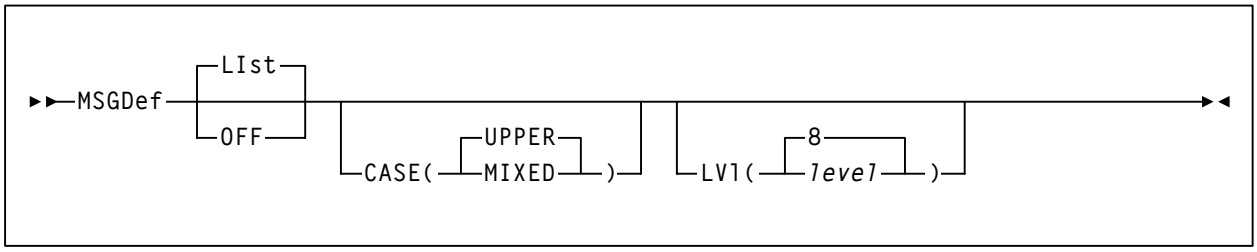
In the following example, the LIst command displays the given length (FFF) of accessible storage beginning with address 01FE00.

```
LIST A=Ø1FEØØ L=FFF
```

MSGDef Command

The SMC MSGDef command defines the appearance of SMC system messages, and controls which messages are displayed and suppressed.

Syntax



Command Name

MSGDef
initiates the MSGDef command.

Parameter Descriptions

LIst
optionally, lists current default SMC message settings.

This is the default if no parameters are specified. It **cannot** be specified with any other parameter.

OFF
optionally, resets all MSGDef values to original SMC default settings. It **cannot** be specified with any other parameter.

CASE
optionally, specifies the message case. Valid values are UPPER or MIXED. If no value is specified, the default value UPPER is used.

LVl
optionally, specifies the default level used to control which SMC messages are displayed and suppressed.

level
indicates the default level. Valid values include the following:

0	Display error messages **only**.
4	Display error and warning messages from the SMC subsystem.
8	Display all SMC subsystem messages and allocation job log warning messages. This is the default value if the MSGDef parameter is not specified.

If no value is specified, the default value of 8 is used.

Note: Levels higher than 8 are used for diagnostic purposes and should **only** be specified as directed by StorageTek Software Support.

Example

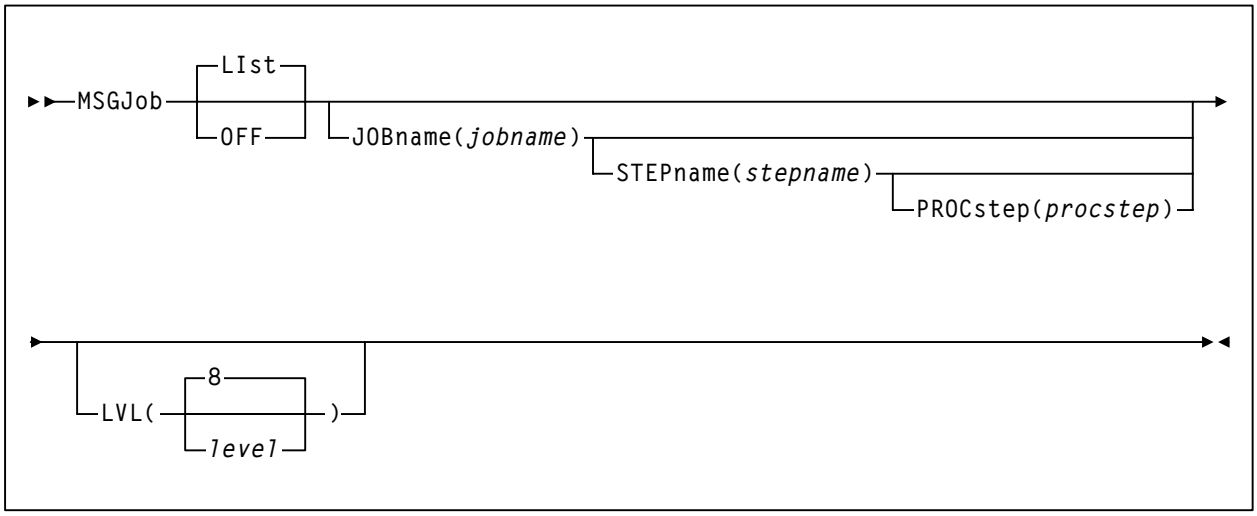
In the following example, the MSGDef command specifies that SMC messages are displayed in mixed case. The command also specifies that **only** error and warning messages from the SMC subsystem are displayed.

```
MSGDEF CASE(MIXED) LVL(4)
```

MSGJob Command

The SMC MSGJob command allows an override of the default message level by job name, step name, and PROC step.

Syntax



Command Name

MSGJob
initiates the MSGJob command.

Parameter Descriptions

LIst
optionally, lists current default SMC message settings and job name, step name, and PROC step exceptions (in the order of evaluation).

This is the default if no parameters are specified. It **cannot** be specified with any other parameter.

OFF
optionally, removes MSGJob overrides.

- If no other parameters are specified, MSGJob overrides are removed for all jobs.
- If only JOBname is specified, MSGJob overrides are removed for ALL MSGJob entries for that job name.
- If JOBname, STEPname and PROCstep are specified, MSGJob overrides are removed **only** for the specified entry.

The global MSGJob is not affected by this parameter.

JOBname
optionally, specifies a job name.

jobname
indicates the job name.

The value entered **must** be one to eight characters in length. An asterisk (*) can be used as a wildcard when included as the last character. In this case, the message level is set for all jobs whose job names match the characters preceding the asterisk.

STEPname
optionally, specifies a step name. It is **only** valid when JOBname is also specified (may be "JOB=*").

stepname
indicates the step name.

The value entered **must** be one to eight characters in length. This parameter is **only** valid if JOBname is also specified.

PROCstep
optionally, specifies a PROC step. It is **only** valid when JOBname and STEPname are also specified. It is **not** valid in JES3 environments with TAPE SETUP processing.

procstep
indicates the PROC step.

The value entered **must** be one to eight characters in length. This parameter is **only** valid if JOBname and STEPname are also specified.

LVL
optionally, specifies the default level used to control which SMC messages are displayed. This parameter is required when JOBname is specified.

level
indicates the default level. Valid values include the following:

0	Display error messages **only**.
4	Display error and warning messages from the SMC subsystem.
8	Display all SMC subsystem messages and allocation job log warning messages.

If no value is specified, the default value of 8 is used.

Note: Levels higher than 8 are used for diagnostic purposes and should **only** be specified as directed by StorageTek Software Support.

Example

In the following example, the MSGJob command specifies that **only** error and warning messages are displayed for job PRODJOB1, step STEP3, PROC step REPORT.

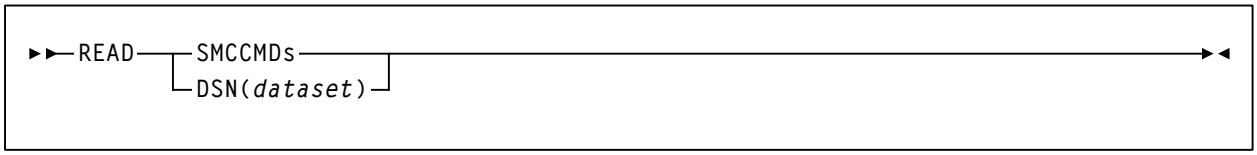
```
MSGJOB JOB(PRODJOB1) STEP(STEP3) PROC(REPORT) LVL(4)
```

READ Command

The SMC READ command is used to enter a series of commands using an input data set instead of console commands.

Note: This command can **only** be issued from the console.

Syntax



Command Name

READ
initiates the READ command.

Parameter Descriptions

SMCCMDs
re-processes commands contained in the data set specified in the SMCCMDS DD statement of the SMC START procedure.

DSN
processes commands contained in the specified data set.

dataset
indicates the specified data set.

Note: If the DSN parameter specifies a member name, the full data set **must** be enclosed in single quotes.

Example

In the following example, the READ command is used to process commands included in the MYCMDS member of the MY.COMMAND.PDS data set.

```
READ DSN('MY.COMMAND.PDS(MYCMDS)')
```

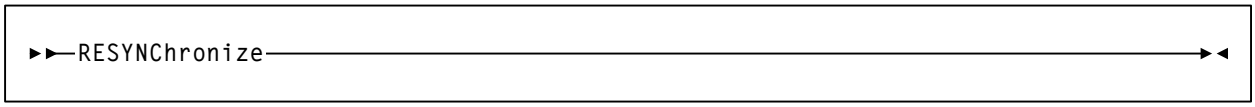
RESYNChronize Command

The SMC RESYNChronize command tests all system drives to determine the library subsystem owner and drive type for each drive.

This action is automatically performed when the SMC detects that a library subsystem has been started or stopped. However, the operator **must** issue the RESYNChronize command in the following instances:

When MVS/CSC is active, issue the SMC RESYNChronize command whenever the MVS/CSC RESYNC command is issued.

Syntax



Command Name

RESYNChronize
initiates the RESYNChronize command.

Parameter Descriptions

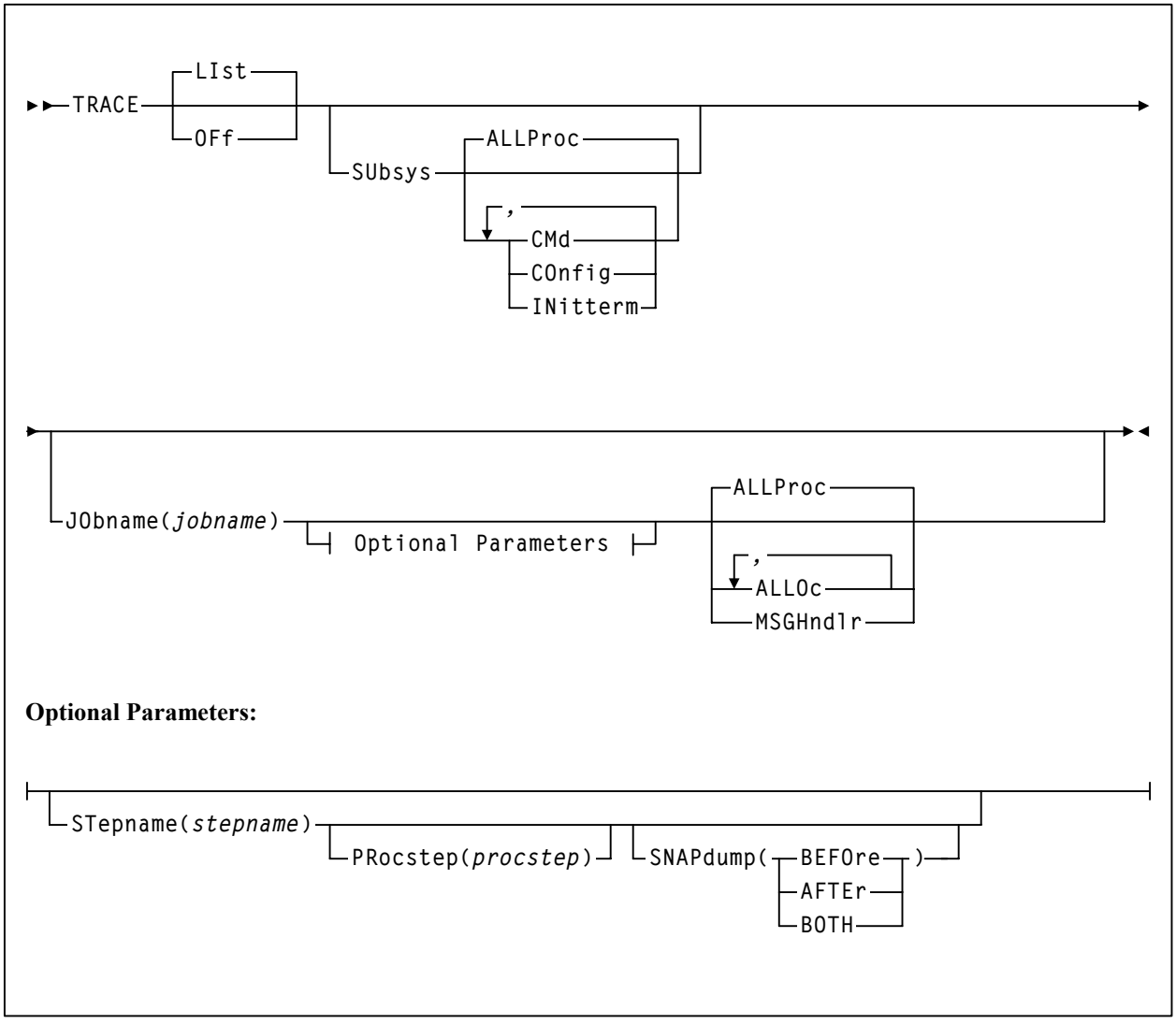
None.

TRace command

The SMC TRace command enables SMC tracing using the GTF trace facility and optionally provides a snap dump of the address space before and after control block modification.

Warning: This command may impact system performance and should be used **only** as directed by StorageTek Software Support.

Syntax



Command Name

TRace
initiates the TRace command.

Parameter Descriptions

LIst
optionally, lists current SMC trace settings for both the SMC subsystems and SMC components by job name, step name, and PROC step.

If no JObname is specified, all jobs with SMC components being traced are listed. If JObname is specified, **only** those jobs with SMC components being traced whose job name matches the specified job JObname are listed. In either case, all SMC subsystems being traced are listed.

This is the default if no parameters are specified.

OFf
optionally, disables SMC tracing.

- If no other parameters are specified, tracing is disabled for all jobs.
- If only JObname is specified, tracing is disabled for all entries for that job name.
- If JObname, STepname and PRocstep are specified, tracing is disabled **only** for the specified entry.

SUbsys
optionally, enables or disables SMC tracing for one or more SMC subtasks.

ALLProc
enables tracing for all SMC subtasks.

OFf
disables tracing for all SMC subtasks.

CMd
enables tracing for the SMC subtask that processes all SMC operator commands.

COnfig
enables tracing for the SMC subtask that maintains information about the environment's tape configuration (i.e. available tape devices and library subsystems (MVS/HSC and MVS/CSC).

INitterm
enables tracing for the SMC subtask that performs SMC initialization and termination.

JObname
optionally, specifies a job name.

jobname
indicates the job name.

The value entered **must** be one to eight characters in length. An asterisk (*) can be used as a wildcard when included as the last character. In this case, all jobs whose job names match the characters preceding the asterisk are traced or listed. To trace all jobs, specify TRACE JOB(*).

STepname
optionally, specifies a step name.

stepname
indicates the step name.

The value entered must be one to eight characters in length. This parameter is **only** valid if JOBname is also specified. It is **not** valid in JES3 environments with TAPE SETUP processing.

PRocstep
optionally, specifies a PROC step.

procstep
indicates the PROC step.

The value entered **must** be one to eight characters in length. This parameter is **only** valid if JObname and STepname are also specified. It is **not** valid in JES3 environments with TAPE SETUP processing.

SNAPdump
optionally, requests a snap dump. This parameter is **not** valid in JES3 environments with TAPE SETUP processing.

Note: A snap dump may impact system performance, and should be used **only** as directed by StorageTek Software Support.

BEFORe
perform a snap dump before allocation influencing.

AFTEr
perform a snap dump after allocation influencing.

BOTH
perform a snap dump both before and after allocation influencing.

ALLProc
enables tracing for all SMC components. This is the default if no parameters are specified.

ALLOc
optionally, enables tracing for the SMC allocation component that performs drive exclusion and drive prioritization.

MSGHndlr
optionally, enables tracing for the SMC message handler component that processes all MOUNT, DISMOUNT, and SWAP messages and issues robotic movement commands to the StorageTek tape library.

Examples

In the following example, tracing is enabled for any job whose name begins with "TESTJOB".

```
TRACE JOB=TESTJOB*
```

In the following example, the TRace command specifies that tracing is disabled for all SMC components and subsystems.

```
TR OFF
```

In the following example, the tracing is enabled for SMC allocation and message handling components for any job with jobname=ABC.

```
TR JOBN(ABC) ALLOC MSGHN
```

In the following example, tracing is enabled for SMC configuration and command subsystems.

```
TR SUBSYS CONFIG CMD
```

In the following example, tracing is disabled for all SMC subsystems.

```
TR SUBSYS OFF
```

In the following example, tracing is disabled for all jobs whose job name begins with "PROD".

```
TR OFF JOB(PROD*)
```

USERMsg Command

The USERMsg command allows you to specify additional message identifiers for messages to be intercepted and passed to the library subsystem's user exit 01. This command can also be used to list the message identifiers of those messages that have been defined using the USERMsg command.

Syntax

Command Name

USERMsg
initiates the USERMsg command.

Parameter Descriptions

LIst
optionally, lists the message identifiers in the customer-defined message table. This is the default if no parameters are specified.

ID
optionally, specifies a message to be added to the customer-defined message table. This message is passed to any active library subsystem user exit 01 (SLSUX01/SCSUX01).

Note: This parameter may **only** be specified in the SMCPARMS data set.

msgid
indicates the message identifier for the message to be added.

Example

In the following example, the USERMsg command is used to list message identifiers for messages included in the customer-defined message table.

```
USERMsg LI
```

Chapter 7. Recovery Procedures

Overview

This chapter describes recovery procedures used when the SMC, a library subsystem (HSC or MVS/CSC), or JES3 becomes inactive. Procedures for both JES2 and JES3 are included.

SMC Recovery Procedures (JES2)

This section describes recovery procedures for the following problem scenarios:

- Inactive SMC - Active library subsystem
- Active SMC - Inactive library subsystem

Inactive SMC - Active Library Subsystem

When the SMC fails while a library subsystem remains active, the following functions are **not** performed:

- allocation processing
- automation of mount/dismount/swap messages

When this occurs, re-start the SMC.

Certain software products allow you to suspend processing that may require dynamic allocation. For example, if Data Facility Hierarchical Storage Manager (DFHSM) is installed on the local processor, you can issue commands to prevent this type of processing without stopping DFHSM.

Common allocations can be postponed by holding the job queue or purging all initiators. Refer to the appropriate IBM publication for more information about JES2 operator commands.

If the HSC and MVS/CSC were started with the AMPND parameter, outstanding mount messages can be re-driven when SMC is re-started. See the *MVS/CSC Configuration Guide* for more information about the AMPND parameter.

Active SMC - Inactive Library Subsystem

When a library subsystem fails or is terminated, volumes and drives owned by that subsystem become unknown to SMC. The following functions are **not** performed:

- allocation processing
- automated mount processing

When this occurs, re-start the library subsystem.

Certain software products allow you to suspend processing that may require dynamic allocation. For example, if Data Facility Hierarchical Storage Manager (DFHSM) is installed on the local processor, you can issue commands to prevent this type of processing without stopping DFHSM.

Common allocations can be postponed by holding the job queue or purging all initiators. Refer to the appropriate IBM publication for more information about JES2 operator commands.

SMC Recovery Procedures (JES3)

This section describes recovery procedures for the following problem scenarios:

- Inactive SMC - Active library subsystem
- Active SMC - Inactive library subsystem
- Inactive JES3 on a local processor
- Inactive JES3 on a global processor

Inactive SMC - Active Library Subsystem

When the SMC fails while a library subsystem (HSC and/or MVS/CSC) remains active, the following functions are **not** performed:

- allocation processing
- automation of mount/dismount/swap messages

When this occurs, re-start the SMC.

Certain software products allow you to suspend processing that may require dynamic allocation. For example, if Data Facility Hierarchical Storage Manager (DFHSM) is installed on the local processor, you can issue commands to prevent this type of processing without stopping DFHSM.

To postpone the C/I process for batch jobs while SMC is inactive, use the following modify command:

```
*F X,D=POSTSCAN,MC=ØØ
```

After the SMC is re-started, restore the maximum count to its original value, xx:

```
*F X,D=POSTSCAN,MC=xx
```

If the HSC and MVS/CSC were started with the AMPND parameter, outstanding mount messages can be re-driven when SMC is re-started. See the *MVS/CSC Configuration Guide* for more information about the AMPND parameter.

Active SMC - Inactive Library Subsystem

When a library subsystem fails or is terminated, volumes and drives owned by that subsystem become unknown to SMC. The following functions are **not** performed:

- allocation processing
- automated mount processing

When this occurs, re-start the library subsystem.

Certain software products allow you to suspend processing that may require dynamic allocation. For example, if Data Facility Hierarchical Storage Manager (DFHSM) is installed on the local processor, you can issue commands to prevent this type of processing without stopping DFHSM.

Inactive JES3 on a Local Processor

When JES3 fails on a local processor, jobs executing at the time that do not require JES3 services continue to execute. Drive exclusion still occurs for dynamic allocation requests.

To recover, restart JES3 (LOCAL start).

The SMC continues processing and requires no recovery.

Inactive JES3 on a Global Processor

When JES3 fails on a global processor, jobs executing at the time that do not require JES3 services continue to execute. Drive exclusion still occurs for dynamic allocation requests.

To recover, restart JES3 or invoke Dynamic System Interchange (DSI) processing.

You can use DSI to reassign the JES3 global function to a JES3 local processor when the global processor becomes inactive or requires maintenance. One of the JES3 local processors becomes the new JES3 global processor. By reassigning the global function to a local processor, the JES3 environment continues processing. The SMC continues processing and requires no recovery.

See the *HSC System Programmer's Guide* or *MVS/CSC System Programmer's Guide* for more information about cross host recovery.

Chapter 8. JES3 Configuration Report Utility

Overview

The SMC utility SMCUPJS generates JES3 configuration data which can be used to help verify that the JES3 initialization deck defines the intended devices and esoterics. This data is derived from internal SMC JES3 configuration control structures. These control structures are built during SMC/JES3 initialization through analysis of the SETNAME and SETUNIT statements in the user's JES3 initialization deck. For this reason, the SMC must be active on the system on which the report is run.

The following reports are generated from the SMC JES3 configuration control structures:

- A list of XTYPE numbers in hexadecimal order with their corresponding XTYPE names.
- A list of XTYPEs in hexadecimal order with all esoterics in which the XTYPE appears.
- A list of esoterics in alphabetical order with all XTYPEs that each contains and the total number of devices in the esoteric. This report also lists any discrepancies between JES3 and HCD esoterics.
- A list of XTYPEs in hexadecimal order with all devices in that XTYPE, along with the location and recording technique of the devices in the XTYPE.
- A list of devices in hexadecimal order and the XTYPE to which each belongs, along with the location and recording technique of the device.

Sample JCL

The following JCL sample executes the SMCUPJS utility to produce the five individual reports listed above.

```
//yourstd JOB card
//STEP EXEC PGM=SMCUPJS
//STEPLIB DD DSN=yoursmc.LINKLIB,DISP=SHR
//STDOUT  DD DSN=yourstd.printout.dataset,DISP=SHR
```

Figure 1. JCL to Produce the JES3 Configuration Report

Note: *yourstd.printout.dataset* should be allocated with LRECL=80.

Sample Reports

In the following report fragments, XTYPE 27 is carried through all five reports to illustrate how the reports are related. Lines pertaining to XTYPE 27 are underlined in each report.

XTYPE Number to XTYPE Name (EXTON)

Below are fragments of the XTYPE Number to XTYPE Name report. This mapping is useful when reading other reports that reference the XTYPE number instead of the name. The number in parenthesis is the XTYPE number, followed by the XTYPE name.

```
SMCUPJS (5.1.0)       SMC JES3 CONFIGURATION REPORT              PAGE 0001
TIME 15:03:42       XTYPE NUMBER TO XTYPE NAME (EXTON)      DATE 2002-01-09

  XTYPE(01) DEV01D5
  XTYPE(02) DEV2504
  XTYPE(03) DEV2507
  XTYPE(04) DEV2506
  XTYPE(05) DEV01D6
  XTYPE(06) DEV01DA
  XTYPE(07) DEV01DD
  XTYPE(08) DEV01DF
  XTYPE(09) DEV01D0
...
  XTYPE(27) DEV0AA0
  XTYPE(28) DEV0A20
...
```

XTYPE Number to Esoteric (EXTOE)

Below are fragments of the XTYPE Number to Esoteric report, which lists each XTYPE in hexadecimal order, along with all esoterics that contain that XTYPE. Both the XTYPE number (in parenthesis) and name are displayed.

```
SMCUPJS (5.1.0)   SMC JES3 CONFIGURATION REPORT                    PAGE 0001
TIME 15:03:42     XTYPE NUMBER TO ESOTERIC (EXTOE)           DATE 2002-01-09


  XTYPE(01) DEV01D5  appears in 11 ESOTERICs
    CART     EAGLE    3590-1   HVALL   HSCV3590 MCZZZZ0   HVZZZZ01 MCZZ9840
    HV994035 HRACS1L1 HV994010
  XTYPE(02) DEV2504  appears in 10 ESOTERICs
    CART
...
  XTYPE(27) DEV0AA0  appears in 7 ESOTERICs
    SYS3480R  CART     3480     TL3480   SLIB     SACS0   S04480
```

Esoteric to XTYPE (EETOX)

Below are fragments of the Esoteric to XTYPE report, which lists each esoteric in alphabetical order, along with all XTYPES in that esoteric in hexadecimal order. The XTYPE number can be mapped to its corresponding XTYPE name using the XTYPE Name to XTYPE Number report described above.

```
SMCUPJS (5.1.0)       SMC JES3 CONFIGURATION SUPPORT                PAGE 0001
TIME 15:03:42        ESOTERIC TO XTYPE NUMBER (EETOX)         DATE 2002-01-09

  ESOTERIC AUSALL   contains 4 XTYPEs and 12 devices
    2C 2D 3D 85
  ESOTERIC AUSREDW  contains 1 XTYPEs and 2 devices
    2D
...
  ESOTERIC CART     contains 84 XTYPEs and 433 devices
    01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 21 22 23 24 25 26 27
    28 29 2A 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 40 41 42
    43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 51 52 53 54 55 56 57 58 59 5A
    5B 5C 5D 5E 5F 60 61 62 63 64 65 66
...
  ESOTERIC SACS0    contains 9 XTYPEs and 22 devices
    10 11 27 42 43 44 45 5E 5F
SMCU0005 *** WARNING: HCD esoteric does not match JES3 esoteric
  JES3 esoteric drives not in HCD:
    0AA3
  HCD esoteric drives not in JES3:
    (NONE)
...
  ESOTERIC SLIB       contains 13 XTYPEs and 34 devices
    10 11 23 24 27 42 43 44 45 46 47 5E 5F
...
  ESOTERIC SYS3480R contains 102 XTYPEs and 1511 devices
    21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38
    39 3A 3B 3C 3D 3E 3F 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50
    51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F 60 61 62 63 64 65 66 67 68
    69 6A 6B 6C 6D 6E 6F 70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F 80
    81 82 83 84 85 86
...
  ESOTERIC S04480  contains 1 XTYPEs and 4 devices
    27
...
  ESOTERIC TL3480  contains 5 XTYPEs and 48 devices
    25 26 27 28 29
...
  ESOTERIC 3480     contains 14 XTYPEs and 132 devices
    21 22 23 24 25 26 27 28 29 83 84 85 86
...
```

XTYPE to Device Number (EXTOD)

Below are fragments of the XTYPE to Device Number report, which lists each XTYPE in hexadecimal order along with its devices. For each XTYPE number the following information is displayed:

- XTYPE number
- XTYPE name
- number of devices in the XTYPE
- associated library subsystem
- device location
- device recording technique.

The device location is one of the following:

- in a library, in which case "SSSS LSM AA:L" is displayed
- in a VTSS, in which case "SSSS VTSS (*nnnnnnnn*)" is displayed
- in the nonlibrary esoteric, in which case "nonlib" is displayed
- neither in the library or nonlibrary esoteric, in which case "not lib or nonlib" is displayed

The recording technique for non-virtual devices is displayed using the same values used in the RECTECH parameter of the HSC UNITATTR control statement. For devices outside the library, if no UNITATTR statement was defined by the HSC, the recording technique is defaulted according to the device type: 3480 defaults to 18TRACK, 3490 defaults to 36ATRACK, and 3590 defaults to STK1RA35.

```
SMCUPJS (5.1.Ø)          SMC JES3 CONFIGURATION REPORT              PAGE ØØØ1
TIME 15:Ø3:42            XTYPE TO DEVICE NUMBER (EXTOD)       DATE 2ØØ2-Ø1-Ø9

  XTYPE(Ø1) DEVØ1D5  contains 1 DEVICEs  not lib or nonlib, rectech STK1RA35
    Ø1D5
  XTYPE(Ø2) DEV25Ø4  contains 1 DEVICEs  not lib or nonlib, rectech STK1RA35
    25Ø5
...
  XTYPE(ØF) DEV28Ø1  contains 1 DEVICEs  nonlib, rectech STK1RA35
    28Ø1
...
  XTYPE(27) DEVØAAØ  contains 4 DEVICEs  HSCQ LSM ØØ:Ø, rectech 18TRACK
    ØAAØ ØAA1 ØAA2 ØAA3
...
  XTYPE(48) DEV9ØØØ  contains 64 DEVICEs  HSCQ VTSS SVTSS1
    9ØØØ 9ØØ1 9ØØ2 9ØØ3 9ØØ4 9ØØ5 9ØØ6 9ØØ7 9ØØ8 9ØØ9 9ØØA 9ØØB 9ØØC 9ØØD
    9ØØE 9ØØF 9Ø1Ø 9Ø11 9Ø12 9Ø13 9Ø14 9Ø15 9Ø16 9Ø17 9Ø18 9Ø19 9Ø1A 9Ø1B
    9Ø1C 9Ø1D 9Ø1E 9Ø1F 9Ø2Ø 9Ø21 9Ø22 9Ø23 9Ø24 9Ø25 9Ø26 9Ø27 9Ø28 9Ø29
    9Ø2A 9Ø2B 9Ø2C 9Ø2D 9Ø2E 9Ø2F 9Ø3Ø 9Ø31 9Ø32 9Ø33 9Ø34 9Ø35 9Ø36 9Ø37
    9Ø38 9Ø39 9Ø3A 9Ø3B 9Ø3C 9Ø3D 9Ø3E 9Ø3F
...
```

Device Number to XTYPE (EDTOX)

Below are fragments of an Device Number to XTYPE report, which lists each device in hexadecimal order and the XTYPE to which it belongs. For each device the following is displayed:

- associated library subsystem
- device location
- device recording technique.

The device location and recording technique are displayed as in the XTYPE to Device Number report described above.

```
SMCUPJS (5.1.0)          SMC JES3 CONFIGURATION REPORT              PAGE 0001
TIME 15:03:42            DEVICE TO XTYPE NUMBER (EDTOX)       DATE 2002-01-09

  DEVICE/XTYPE - 0120/3C  not lib or nonlib, rectech 36ATRACK
  DEVICE/XTYPE - 0121/3C  not lib or nonlib, rectech 36ATRACK
...
  DEVICE/XTYPE - 0AA0/27  HSCQ LSM 00:0, rectech 18TRACK
  DEVICE/XTYPE - 0AA1/27  HSCQ LSM 00:0, rectech 18TRACK
  DEVICE/XTYPE - 0AA2/27  HSCQ LSM 00:0, rectech 18TRACK
  DEVICE/XTYPE - 0AA3/27  HSCQ LSM 00:0, rectech 18TRACK
...
```

JES3 Configuration Utility Return Codes and Messages

The JES3 configuration utility report program may return the following return codes:

- A return code of 0 is returned when the configuration utility report program completes successfully with no errors or warnings.
- A return code of 4 is returned when an inconsistency in the JES3 configuration is detected.
- A return code of 8 is returned when an error that prevents production of the report is detected.

For a listing of SMC messages that may be returned by the JES3 utility report program, see page 178. These messages are identified by the `SMCU` prefix.

Chapter 9. SMC Messages

Overview

This chapter provides information about SMC messages. This information is provided to help system programmers and operators:

- initialize the SMC
- monitor SMC activity
- diagnose and correct SMC problems

Message Format

Messages are displayed on the console in the following format:

`SMC`*nnnn* *message-text*

where:

- `SMC` identifies the Storage Management Component (SMC).
- *nnnn* is the four-character message number.
- *message-text* is the actual text displayed on the job log or system log.

Message Descriptions

This chapter provides a description for each SMC message. In addition to the message number and message text, this description includes the following:

Message Level

The message level represents a message category. Using the MSGDEF or MSGJOB operator command, the message level is specified in order to control which categories of messages are issued. See "MSGDef Command" on page 123 for more information.

Explanation

The explanation describes the message.

System Action

The system action describes how the SMC reacts when a message-triggering event occurs.

User Response

The user response describes how the user should respond to the message. In many cases, no response is required.

Variable Definitions

Italicized text indicates variable data that is replaced by actual values when messages are issued. Message specific variable data is symbolized as follows:

Table 1. Variable Data Definitions

Variable Data	Definition
AA	ACSid
n	decimal value
X	hexadecimal value
various letters (*C*, *N*, *K*, etc.)	variable information (character data)
numbers (*1*, *2*, etc.) (combined with letters)	related variable information i.e. commands *CCCCCCCC1* and *CCCCCCCC2*
{ }	multiple choices
[ ]	optional field (may not appear in message)

Message Listing

The following SMC messages are listed numerically.

SMC0001 `SMC subsystem initializing`

Message Level: 0

Explanation: The MVS start command was entered for the SMC, and the SMC subsystem initialization process has begun.

System Action: None.

User Response: None.

SMC0002 *`CCCCCCCCCCCCCCC`* `failed; return code=`*`XXXX`*`, reason code=`*`XXXX`*

Message Level: 0

Explanation: MVS facility or macro *CCCCCCCCCCCCCCCC* completed with the specified non-zero return code *XXXX* and reason code *XXXX*.

System Action: Depending upon the type of error, initiation/termination may try to continue. If the MVS facility is listed as *Requested SDUMP*, the error occurred during the TRACE SNAP process, and processing will continue without producing the requested SDUMP.

User Response: Look for IBM related messages in the SYSLOG or job log, and refer to IBM documentation for the explanation.

SMC0003 `SMC subsystem` *`SSSS`* `terminating`

Message Level: 0

Explanation: The MVS stop command was entered for the SMC, and the SMC subsystem termination process has begun.

System Action: None.

User Response: None.

SMC0004 `MVS release is down-level`

Message Level: 0

Explanation: An attempt was made to initialize the SMC on an MVS system that does not support the necessary services required by this version of SMC.

System Action: The SMC subsystem terminates.

User Response: Upgrade MVS to the required release level.

SMC0005 `Invalid command CCCCCCCC [at line nnnn of SMCCMDS|SMCPARMS]`

Message Level: 0

Explanation: An undefined command, *CCCCCCCC*, was encountered by the SMC.

System Action: Processing continues. The command is ignored.

User Response: Ensure that the syntax in the command data set is correct, or enter the corrected command.

SMC0006 `Subsystem not dynamic`

Message Level: 0

Explanation: The SMC was defined as a non-dynamic subsystem using the positional form of the IEFSSN*xx* parmlib member. The SMC must run as a dynamic subsystem.

System Action: The SMC subsystem terminates.

User Response: Use the keyword format of the subsystem definition in the IEFSSN*xx* parmlib member to define the SMC subsystem, or remove the SMC subsystem from the IEFSSN*xx* parmlib member.

SMC0007 `SMCCVT incompatible with previous version; defaulting to COLD start`

Message Level: 4

Explanation: During initialization, the SMC subsystem detected that the prior SMC subsystem of the same name was an incompatible version. The COLD start parameter was not specified.

System Action: Initialization continues, but in COLD start mode (the SMC subsystem CVT is rebuilt).

User Response: None.

SMC0008 `Not running from an authorized library; subsystem terminating`

Message Level: 0

Explanation: The SMC subsystem initialization module, SMCBINT, was executed from an unauthorized library.

System Action: The SMC subsystem terminates.

User Response: Ensure that the SMC link library is APF authorized.

SMC0009 `Job JJJJJJJJ JOBnnnn active at termination`

Message Level: 4

Explanation: The SMC subsystem received the MVS stop ('P') command, but there is an active job, *JJJJJJJJ*, in tape allocation. The SMC0012 message was issued previously, but the job allocation has not completed after waiting one minute.

System Action: Termination continues. No allocation influencing is performed for JOB*nnnn*.

User Response: None.

SMC0010 `Unable to acquire storage for CCCCCCCC; return code=XXXX`

Message Level: 0

Explanation: During initialization, the SMC subsystem could not acquire sufficient storage for the specified dynamic control block or module, *CCCCCCCC*.

System Action: The SMC subsystem terminates.

User Response: Ensure that there is sufficient CSA storage available. Refer to IBM documentation for the explanation of return code *XXXX*.

SMC0011 `Load failed for module MMMMMMMM`

Message Level: 0

Explanation: The SMC subsystem could not load the required module *MMMMMMMM*.

System Action: The SMC subsystem terminates.

User Response: Ensure that the SMC startup procedure has access to all SMC distributed load libraries in its steplib concatenation.

SMC0012 `Termination waiting for job JJJJJJJJ JOBnnnnn [in allocation|in message handling]`

Message Level: 0

Explanation: The SMC subsystem received the MVS stop command, but there is an active job *JJJJJJJJ* in tape allocation or message handling.

System Action: The SMC waits for 30 seconds or until all active processes are complete.

User Response: Ensure that there are no jobs performing tape allocation or message handling when the SMC is terminated. Respond to any MVS allocation recovery messages.

SMC0013 `TRACE SUBSYS SSSSSSSS`

Message Level: 0

Explanation: The TRACE command was specified with the LIST keyword. The SMC0013 message lists the SMC subsystems that are traced.

System Action: None.

User Response: None.

SMC0014 `Unmatched quote detected; command ignored [at line nnnn of SMCCMDS|SMCPARMS]`

Message Level: 0

Explanation: The SMC encountered a command containing an unterminated quoted string.

System Action: Processing continues. The command is ignored.

User Response: Ensure that the syntax in the command data set is correct, or enter the corrected command.

SMC0015 `Invalid keyword KKKKKKKK for the CCCCCCCC command [at line nnnn of SMCCMDS|SMCPARMS]`

Message Level: 0

Explanation: The SMC encountered a command that specified an invalid keyword *KKKKKKKK*.

System Action: Processing continues. The command is ignored.

User Response: Ensure that the syntax in the command data set is correct, or enter the corrected command.

SMC0016 `Invalid value VVVVVVVV for keyword KKKKKKKK of the CCCCCCCC command [at line nnnn of SMCCMDS|SMCPARMS]`

Message Level: 0

Explanation: The SMC encountered a command that specified keyword *KKKKKKKK* with an invalid value *VVVVVVVV*.

System Action: Processing continues. The command is ignored.

User Response: Ensure that the syntax in the command data set is correct, or enter the corrected command.

SMC0017 Keyword *KKKKKKKK* of the *CCCCCCCC* command requires a value [at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The SMC encountered a command that specified keyword *KKKKKKKK* without an accompanying value (required by most keywords).

System Action: Processing continues. The command is ignored.

User Response: Ensure that the syntax in the command data set is correct, or enter the corrected command.

SMC0018 Keyword *KKKKKKKK* of the *CCCCCCCC* command is not allowed for *EEEEEEEE* [at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The SMC encountered a command that specified keyword *KKKKKKKK,* which is not valid in the current operating environment *EEEEEEEE*. For example, some keywords or keyword=value pairs may be invalid depending upon whether the user is executing JES2 or JES3.

System Action: Processing continues. The command is ignored.

User Response: Ensure that the specified keyword is valid in your environment.

SMC0019 Duplicate keyword *KKKKKKKK* specified for the *CCCCCCCC* command [at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The SMC encountered a command that specified the same keyword, *KKKKKKKK*, more than once.

System Action: Processing continues. The command is ignored.

User Response: Ensure that the syntax in the command data set is correct, or enter the corrected command.

SMC0020 `Keyword KKKKKKK1 of the CCCCCCCC command is mutually exclusive with keyword KKKKKKK2 [at line nnnn of SMCCMDS|SMCPARMS]`

Message Level: 0

Explanation: The SMC encountered a command that specified multiple keywords, two of which (*KKKKKKK1* and *KKKKKKK2)*, are mutually exclusive.

System Action: Processing continues. The command is ignored.

User Response: Ensure that the syntax in the command data set is correct, or enter the corrected command.

SMC0021 `{COLD|WARM} start failure`

Message Level: 0

Explanation: During initialization, the SMC subsystem detected an error.

System Action: The SMC subsystem terminates.

User Response: Look for SMC related messages in the SYSLOG or job log. Associated messages may be (but are not limited to) SMC0002, SMC0004, SMC0006, SMC0008, SMC0010, or SMC0011.

SMC0022 `Invalid format for the CCCCCCCC command [at line nnnn of SMCCMDS|SMCPARMS]`

Message Level: 0

Explanation: The SMC encountered a command *CCCCCCCC* that contained either too many or too few keywords in the command line.

System Action: Processing continues. The command is ignored.

User Response: Ensure that the syntax in the command data set is correct, or enter the corrected command.

SMC0023 `CCCCCCCC command successfully processed [at line nnnn of SMCCMDS|SMCPARMS]`

Message Level: 0

Explanation: The *CCCCCCCC* command was successfully validated and processed by the SMC.

System Action: None.

User Response: None.

SMC0024 *SSSS* subsystem initialization complete

Message Level: 0

Explanation: The SMC initialization process is complete. The SMC is ready to begin normal operations.

System Action: None.

User Response: None.

SMC0025 No {ALLOCJOB|MSGJOB|TRACE JOB|TRACE SUBSYS|control block} entries to display [(no JES3 tape setup)]

Message Level: 0

Explanation:

One of the following occurred:

- An ALLOCJOB, MSGJOB, or TRACE command was specified with the LIST keyword. However, no entries were found in the SMC queue of ALLOCJOB, MSGJOB, or TRACE commands.
- A LIST command was specified with a control block keyword. However, no control block entries were found.
- A LIST command was specified for an SMC JES3 control block name. However, the system was not operating with JES3 tape setup.

System Action: None.

User Response: None.

SMC0027 Keyword *KKKKKKK1* of the *CCCCCCCC* command requires keyword *KKKKKKK2* [at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The SMC encountered a command that specified keyword *KKKKKKK1*, but not the required co-requisite keyword, *KKKKKKK2*.

System Action: Processing continues. The command is ignored.

User Response: Ensure that the syntax in the command data set is correct, or enter the corrected command.

SMC0028 `STOP command received`

Message Level: 0

Explanation: The MVS stop ('P') command was received by the SMC, and the SMC subsystem termination process is set to begin.

System Action: The SMC subsystem begins termination processing.

User Response: None.

SMC0029 `{ALLOCJOB|MSGJOB|TRACE} command processing error; matching entry not found [at line` *`nnnn`* `of SMCCMDS|SMCPARMS]`

Message Level: 0

Explanation: An ALLOCJOB, MSGJOB, or TRACE command was specified with the "OFF" option, to remove a previously entered ALLOCJOB, MSGJOB, or TRACE job, step, or PROC step. However, the specified entry could not be found in the SMC queue of ALLOCJOB, MSGJOB, or TRACE entries.

System Action: Processing continues. The command is ignored.

User Response: Use the LIST keyword for the ALLOCJOB, MSGJOB, or TRACE command to list the current SMC queue of entries; then ensure that the syntax in the command data set is correct, or enter the corrected command.

SMC0030 `TRACE JOBNAME=`*`JJJJJJJJ`* `[STEPNAME=`*`SSSSSSSS`*`]`
`[PROCSTEP=`*`PPPPPPPP`*`] [SNAPdump` *`DDDDDDDD`*`]`

Message Level: 0

Explanation: The TRACE command was specified with the LIST keyword. The SMC0030 message lists the jobs, step, and PROC steps and the associated processes which have been specified for tracing.

System Action: None.

User Response: None.

SMC0031 `No SMC start mode specified; defaulting to WARM start`

Message Level: 4

Explanation: During SMC initialization, the subsystem detected that neither a WARM or COLD start was specified in the initialization program's execution parameters.

System Action: Initialization continues in WARM start mode.

User Response: None.

SMC0032 `Number of SMC startup parameters specified exceeds maximum of n`

Message Level: 0

Explanation: During initialization, the SMC subsystem initialization program encountered an execution parameter string containing too many parameters.

System Action: The SMC subsystem terminates.

User Response: Correct the subsystem initialization startup procedure to specify the correct execution parameter string.

SMC0033 `SMC startup parameter PPPPPPPP may not have a value`

Message Level: 0

Explanation: During initialization, the SMC subsystem initialization program encountered a valid execution parameter, but it was specified as a keyword=value pair, where no value is allowed.

System Action: The SMC subsystem terminates.

User Response: Correct the subsystem initialization startup procedure to specify the correct execution parameter string.

SMC0034 `SMC startup parameter PPPPPPPP must have a value`

Message Level: 0

Explanation: During initialization, the SMC subsystem initialization program encountered a valid execution parameter, but it was not specified as a keyword=value pair, and a value is required.

System Action: The SMC subsystem terminates.

User Response: Correct the subsystem initialization startup procedure to specify the correct execution parameter string.

SMC0035 `Error processing SMC startup parameter PPPPPPPP; CCCCCCCCCCCC`

Message Level: 0

Explanation: During initialization, the SMC subsystem initialization program encountered an error in the execution parameter string. The string *CCCCCCCCCCCC* indicates the type of error encountered.

System Action: The SMC subsystem terminates.

User Response: Correct the subsystem initialization startup procedure to specify the correct execution parameter string.

SMC0036 `SMC startup parameter PPPPPPPP successfully processed`

Message Level: 4

Explanation: During SMC initialization, the execution parameter *PPPPPPPP* was successfully verified and processed.

System Action: None.

User Response: None.

SMC0037 `Invalid SMC startup parameters; subsystem terminating`

Message Level: 0

Explanation: During initialization, the SMC subsystem initialization program detected an error processing the execution parameter string.

System Action: The SMC subsystem terminates.

User Response: Look for SMC related messages in the SYSLOG or job log. Associated messages may be (but are not limited to) SMC0032 SMC0033, SMC0034, or SMC0035.

SMC0038 `Another SMC system SSSS is already active`

Message Level: 0

Explanation: During initialization, the SMC subsystem initialization program detected a different SMC subsystem, *SSSS,* already active on the system.

System Action: The initializing SMC subsystem terminates.

User Response: Only one SMC can be active on a system at a time.

- If SMC *SSSS* is active, and is preventing the current SMC from initializing, terminate SMC *SSSS*.
- If SMC *SSSS* was terminated abnormally and is not truly active, restart the current SMC subsystem with the RESET execution parameter.

SMC0039 `Identically named subsystem SSSS is already active`

Message Level: 0

Explanation: During initialization, the SMC subsystem initialization program detected an identically named SMC subsystem *SSSS* already active on the system.

System Action: The initializing SMC subsystem terminates.

User Response: Only one SMC can be active on a system at a time. If SMC subsystem *SSSS* was terminated abnormally and is not truly active, then restart SMC *SSSS* with the RESET execution parameter.

SMC0040 `SMC subsystem SSSS is already active; RESET specified; startup continuing`

Message Level: 4

Explanation: During initialization of SMC subsystem *SSSS,* the initialization program detected that the prior SMC subsystem terminated abnormally, but the user specified the RESET execution parameter to ignore this condition.

System Action: None.

User Response: None.

SMC0041 `{Command|Comment} beginning at line nnnn is unterminated`

Message Level: 4

Explanation: A command or comment beginning at line *nnnn* of an input command file ended with a continuation character (+), but no continuation was found.

System Action: Processing continues. The command containing the unterminated string is ignored.

User Response: Ensure that the syntax in the command data set is correct.

SMC0042 `Job JJJJJJJJ step SSSSSSSS not allocatable before SMC modification`

Message Level: 4

Explanation: During execution of job *JJJJJJJJ* step *SSSSSSSS*, the SMC allocation component detected that the job step was not allocatable before any allocation influencing was performed.

System Action: None. The job is failed by MVS or express-canceled by JES3.

User Response: Correct the JCL.

SMC0043 `Job JJJJJJJJ step SSSSSSSS not allocatable at MINLVL=nn; failing DD DDDDDDDD`

Message Level: 4

Explanation: During execution of a job with tape allocation, the SMC allocation component detected that the job is not allocatable at the specified minimum exclusion level (MINLVL).

System Action: The job is failed by MVS or express-canceled by JES3.

User Response: Correct the JCL, or change the MINLVL for the specified job.

SMC0044 `Subsystem termination in progress; no allocation influence for job JJJJJJJJ step SSSSSSSS`

Message Level: 8

Explanation: During execution of job *JJJJJJJJ* step *SSSSSSSS*, the SMC allocation component detected that the SMC subsystem was being terminated.

System Action: No allocation influencing is performed for the job.

User Response: None.

SMC0045 `Conflicting exclusion criteria for job JJJJJJJJ step SSSSSSSS DD DDDDDDDD`

Message Level: 8

Explanation: Messages SMC0045 and SMC0046 are always produced together. See the explanation for message SMC0046 for more information.

System Action: None.

User Response: None.

SMC0046 `CCCCCCCCCCCCCCCC would have excluded all devices; not honored`

Message Level: 8

Explanation: When the SMC allocation component attempted to apply the exclusion criteria *CCCCCCCCCCCCCCCC* for job *JJJJJJJJ* step *SSSSSSSS* DD *DDDDDDDD*, no devices remained in the current exclusion level EDL. The SMC0045 and SMC0046 messages are always produced together.

System Action: None. The criteria that would have eliminated all devices is ignored.

User Response: None.

SMC0047 `Esoteric EEEEEEEE contains no known devices; ignored for job JJJJJJJJ step SSSSSSSS DD DDDDDDDD`

Message Level: 8

Explanation: During execution of job *JJJJJJJJ* step *SSSSSSSS*, the SMC allocation component encountered a user policy esoteric, *EEEEEEEE* (from TAPEREQ or user exit), that did not contain any tape devices in the current EDL.

System Action: The specified esoteric is ignored.

User Response: None.

SMC0048 `ACS ID aa contains no known devices; ignored for job JJJJJJJJ step SSSSSSSS DD DDDDDDDD`

Message Level: 8

Explanation: During execution of job *JJJJJJJJ* step *SSSSSSSS*, the SMC allocation component encountered an user exit policy requesting exclusion to ACS ID *aa*. However, the EDL at the current exclusion level does not include tape devices in the requested ACS.

System Action: The ACS ID policy request is ignored.

User Response: None.

SMC0049 `No eligible HSC or MVS/CSC subsystems for job JJJJJJJJ`

Message Level: 8

Explanation: During execution of job *JJJJJJJJ*, the SMC allocation component did not find an HSC or MVS/CSC subsystem active at the correct release level.

System Action: None. No allocation influencing is performed for the job.

User Response: Ensure that an HSC or MVS/CSC at the correct release level is active on the same host.

SMC0051 `Subsystem SSSS release n.n is not at SMC required level n.n`

Message Level: 8

Explanation: During its attempt to select an HSC or MVS/CSC for allocation or mount processing, SMC discovered an active subsystem at an incorrect release level.

System Action: The indicated subsystem is not used for SMC allocation or mount processing.

User Response: Ensure that HSC and MVS/CSC subsystems on the same MVS host match the required SMC release level.

SMC0052 `User exit nn {inactive|abended and disabled}`

Message Level: 8

Explanation: During execution of a job with tape allocation, the SMC allocation component invoked the user exit *nn* from an HSC or MVS/CSC subsystem. However, the user exit is currently inactive, or has abended and is now disabled for both the SMC and the owning HSC or MVS/CSC subsystems.

System Action: Allocation influencing continues for the job.

User Response: Correct the user exit.

SMC0053 `**** SMC U1Ø99 ABEND at CCCCCCCCn ****`

Message Level: 0

Explanation: An SMC task has abended in module *CCCCCCCC* at abend sequence number *n*.

System Action: If the abend occurs in the address space of a tape allocation job, the SMC subsystem does not influence the job's allocation. If the abend occurs in the SMC started task address space, a restart of the SMC subsystem may be required.

User Response: Look for SMC or IBM related messages in the SYSLOG or job log. Save the associated logs, dump data set, and JCL, and contact StorageTek Technical Support.

SMC0054 `MSGJOB LVL=nn for {DEFAULT SETTING|JOBNAME=JJJJJJJJ} [STEPNAME=SSSSSSSS][PROCSTEP=PPPPPPPP]`

Message Level: 0

Explanation: A MSGJOB command was specified with the LIST keyword. Each unique job, job step, and PROC step entry found in the MSGJOB queue is displayed in a separate SMC0054 message, followed by the SMC defaults in a final SMC0054 message. *nn* indicates the message level, and any messages at that indicated level or lower are produced on that job's job log in JES2 or on the system log in JES3.

System Action: None.

User Response: None.

SMC0055 `ALLOCJOB {BYPASSed|MINLVL=`*`nn`*`}[SEPLVL=MIN|MAX|`*`nn`*`][SMS=OFF|ON] for {JOBNAME=`*`JJJJJJJJ`*`}[STEPNAME=`*`SSSSSSSS`*`][PROCSTEP=`*`PPPPPPPP`*`][SNAPdump `*`DDDDDD`*`] `*`PPPPPPPP`*

Message Level: 0

Explanation: An ALLOCJOB command was specified with the LIST keyword. Each unique job, job step, and PROC step entry found in the ALLOCJOB queue is displayed in a separate SMC0055 message. The BYPASSED message indicates that the specified job, job step, or PROC step will not have any of its tape allocations influenced by the SMC. MINLVL=*nn* indicates the desired minimum exclusion level at which affinity and GDG chains are separated. The SMS value indicates whether the SMC DFSMS interface is enabled for this entry.

System Action: None.

User Response: None.

SMC0056 `Listing `*`nnnn`*` bytes at address `*`XXXXXXXX`*

Message Level: 0

Explanation: The SMC encountered a valid LIST command.

System Action: Following the SMC0056 message, the SMC displays multiple lines listing the storage in translated hexadecimal format, each line listing the next 16 bytes of storage.

User Response: None.

SMC0057 `No {SMCPARMS|SMCCMDS} DDNAME statement found`

Message Level: 8

Explanation: During SMC initialization, the specified SMCPARMS DD or SMCCMDS DD was not present in the SMC startup procedure.

System Action: Initialization continues.

User Response: None.

SMC0058 `Error opening {DDNAME|DSNAME} {SMCPARMS|SMCCMDS|`*`datasetname`*`}`

Message Level: 0

Explanation: The SMC encountered a READ command, but the specified DDNAME or DSNAME could not be opened.

System Action: The READ command is ignored.

User Response: Look for IBM related messages in the SYSLOG or job log, and refer to IBM documentation for more information.

SMC0059 `Identically named subsystem `*`SSSS`*` is initializing`

Message Level: 0

Explanation: During initialization, the SMC subsystem initialization program detected an identically named SMC subsystem, *SSSS*, already being initialized (but not yet fully active).

System Action: The currently initializing SMC subsystem terminates.

User Response: Only one SMC can be active on a system at a time. If the prior SMC subsystem, *SSSS*, was terminated abnormally, and is not truly active, then restart SMC *SSSS* with the RESET execution parameter.

SMC0060 `I/O error reading [DDNAME {SMCPARMS|SMCCMDS}|DSNAME `*`datasetname`*`]`

Message Level: 0

Explanation: The SMC received an I/O error attempting to read the SMCPARMS or SMCCMDS data set specified in the SMC started procedure or a data set specified on a READ command.

System Action: The indicated data set is not processed.

User Response: Determine the cause of the error. If the input data set is a PDS, ensure that a member name was specified.

SMC0061 `Command beginning at line `*`nnnn`*` of {SMCCMDS|SMCPARMS} is too long; input ignored`

Message Level: 4

Explanation: The SMC encountered a multi-line command beginning at line *nnnn* of the specified file. This command exceeds 1024 characters in length.

System Action: Processing continues. The entire multi-line is ignored.

User Response: Ensure that the command data set has the correct syntax.

SMC0062 `Command CCCCCCCC [with parameter PPPPPPPP] is not allowed [{from console|at line nnnn of SMCCMDS|SMCPARMS}]`

Message Level: 0

Explanation: The SMC encountered a command or a command parameter that is not supported for the indicated command origin. For example, the READ command is not supported if encountered during processing of another READ command.

System Action: The command is ignored.

User Response: Issue the command from a valid command origin.

SMC0063 `MSGDEF LVL=nn CASE={UPPER|MIXED}`

Message Level: 0

Explanation: A MSGDEF or MSGJOB command was specified with the LIST keyword. The SMC0063 message lists the default message settings for the SMC subsystem. *nn* indicates the message level, and any messages at that indicated level or lower will be produced. CASE indicates whether messages are converted to mixed case.

System Action: None.

User Response: None.

SMC0066 `ALLOCDEF MINLVL=nn SEPLVL={MIN|MAX|nn} SMS={OFF|ON} MIACOMPAT={OFF|ON}`

Message Level: 0

Explanation: An ALLOCDEF or ALLOCJOB command was specified with the LIST keyword. The SMC0066 message lists the default allocation settings.

- MINLVL indicates the exclusion level at which a job step must have drives remaining in order not to fail allocation.
- SEPLVL indicates the exclusion level at which affinity chains will be separated.
- SMS indicates whether or not the SMC DFSMS interface is enabled.
- MIACOMPAT indicates whether or not the EDL is marked for excluded devices at SSI24 time.

System Action: None.

User Response: None.

SMC0068 `Cannot substitute for esoteric=EEEEEEEE at MINLVL=n job JJJJJJJJ step SSSSSSSS for DD DDDDDDDD`

Message Level: 8

Explanation: During execution of job *JJJJJJJJ* step *SSSSSSSS* on JES3, the SMC allocation component determined that the job step was not allocatable at the specified minimum level (MINLVL).

System Action: The job is express-canceled by JES3.

User Response: None.

SMC0069 `SMCERSLV release level LLLLLL1 does not match SMC release level LLLLLL2`

Message Level: 0

Explanation: Module SMCERSLV is at release level *LLLLLL1* which does not match SMC release level *LLLLLL2*.

System Action: The SMC does not initialize.

User Response: Reassemble SMCERSLV at the correct release level and restart the SMC.

SMC0070 `SMC maintenance has been applied; reassemble SMCERSLV`

Message Level: 0

Explanation: SMC maintenance was applied and affected the module SMCERSLV, but SMCERSLV was not reassembled.

System Action: The SMC does not initialize.

User Response: Reassemble SMCERSLV using the correct level of SMC macros and restart the SMC.

SMC0071 `SMCERSLV JES3 release level LLLLLL1 does not match JES3 release level LLLLLL2`

Message Level: 0

Explanation: Module SMCERSLV was assembled using a different JES3 release level *LLLLLL1* than the active JES3 release level *LLLLLL2*.

System Action: The SMC does not initialize.

User Response: Reassemble SMCERSLV using the correct level of JES3 macros and restart the SMC.

SMC0073 `JES3 C/I waiting for SMC to initialize; Start SMC or reply “GO” to continue`

Message Level: 0

Explanation: JES3 has begun scanning the JCL of jobs requiring tape mounts and the SMC is not initialized and cannot influence allocation.

System Action: One JES3 C/I process waits until the SMC has been started or the operator has replied “GO”.

User Response: Start the SMC or reply “GO” to proceed without SMC allocation influence.

SMC0074 `Unsupported virtual label type for job JJJJJJJJ step SSSSSSSS DD DDDDDDDD`

Message Level: 8

Explanation: All virtual devices were previously excluded for the DD because of an unsupported label type (NL). A subsequent exclusion criteria requesting virtual drives could not be honored. Message SMC0046 describes the criteria not honored.

System Action: The DD is allocated to a non-virtual drive. Non-library drives are preferred over library drives.

User Response: Change the JCL to request a supported label type, or change the policy to direct the allocation to a different device type.

SMC0075 `SEPLVL cannot be less than MINLVL on the CCCCCCCC command [at line nnnn of SMCCMDS|SMCPARMS]`

Message Level: 0

Explanation: The SMC encountered a command that specified either a SEPLVL that is lower than the applicable MINLVL or a MINLVL higher than the applicable SEPLVL.

System Action: Processing continues. The command is ignored.

User Response: Ensure that command MINLVL and SEPLVL values are correct in the command data set, or enter the corrected command.

SMC0076 `Xtype CCCCCCCC (XX) has inconsistent {device type|location type|VTSS|ACS} between device DDDD1 and device DDDD2`

Message Level: 4

Explanation: In a JES3 environment, the SMC encountered an XTYPE that contains non-homogeneous devices, where *CCCCCCCC* represents the JES3 SETNAME name. Use of this XTYPE may result in allocation to an incorrect device based on media, recording technique, or location.

System Action: Allocation is performed based on the device characteristics of the first device encountered in the XTYPE.

User Response: Refer to Chapter 2, "Starting the SMC" for JES3 initialization parameter requirements.

SMC0077 `Subtask {SMCOCMD|SMCFDRV|SMCBMID|SMCJSWP} terminating at retry count n; please start and restart subsystem`

Message Level: 0

Explanation: The specified SMC subsystem task abnormally terminated *N* times, and could not be restarted. The subsystem is now operating without a required service.

System Action: Processing continues. However, specific allocation or command facilities may be affected.

User Response: Stop and restart the SMC subsystem. If the named subtask is SMCOCMD, use the MVS CANCEL command to terminate the subsystem.

SMC0078 `No command prefix defined`

Message Level: 0

Explanation: No CMDDEF command specified a command prefix in the SMCPARMS data set.

System Action: Processing continues.

User Response: If a command prefix is desired, add the CMDDEF command to the SMCPARMS data set. The command is processed the next time SMC is initialized.

SMC0079 `The command prefix is PPPPPPPP`

Message Level: 0

Explanation: The command prefix for the subsystem is set to *PPPPPPPP*.

System Action: The SMC now accepts commands prefixed with *PPPPPPPP*.

User Response: None.

SMC0080 `Command prefix value VVVVVVVV contains invalid character C at line nnnn of SMCPARMS`

Message Level: 0

Explanation: The command prefix value *VVVVVVVV* of the CMDDEF command contained an invalid character *C*.

System Action: Processing continues. The command is ignored.

User Response: Review the list of valid characters for the command prefix listed with the CMDDEF command description. Update the CMDDEF command in the SMCPARMS data set with the new prefix value. The command is processed the next time the SMC is initialized.

SMC0081 `Command prefix not added; [prefix is not unique|CPF system error]`

Message Level: 0

Explanation: The command prefix was disallowed by the CPF facility because the prefix was not unique or a system error occurred.

System Action: Processing continues. The command is ignored.

User Response:

- If the prefix is not unique, issue the MVS DISPLAY OPDATA command and compare the prefix with those of the other subsystems. The command prefix **cannot** include a command string, a command abbreviation, or any string that invokes a command. The command prefix **cannot** include a string that is a subset or superset of an existing prefix beginning with the same character.
- If a CPF error occurred, look for IBM related messages in the SYSLOG and refer to IBM documentation for the explanation.

SMC0082 `Command prefix already set`

Message Level: 0

Explanation: The command prefix can be specified only once while the SMC is initializing and cannot be changed during execution.

System Action: Processing continues. The command is ignored.

User Response: To change the command prefix, update the CMDDEF statement in the SMCPARMS data set and recycle the SMC.

SMC0083 `Unable to locate {JES3 SETNAME table|JES3 SETUNIT table}`

Message Level: 0

Explanation: During SMC subsystem initialization on a JES3 system, the indicated JES3 control structures could not be located. The SMC could not initialize.

System Action: The SMC subsystem terminates.

User Response: Reassemble SMCERSLV with the current level of JES3 macros. Refer to the *NCS Installation Guide* for more information.

SMC0084 *`MMM DD YYYY HH:MM:SS SSSS`* `Active on hostid` *`HHHH`*

Message Level: 4

Explanation: The date (*MMM DD YYYY*), time (*HH*:*MM*:*SS*), subsystem name (*SSSS*) and MVS hostid (*HHHH*) are displayed once a day at midnight and during SMC initialization.

System Action: None.

User Response: None.

SMC0089 `Unable to start subtask [SMCOCMD|SMCFDRV|SMCBMID|SMCJSWP]`

Message Level: 0

Explanation: During initialization, the indicated SMC subsystem task could not be successfully attached.

System Action: The SMC subsystem terminates.

User Response: Look for related MVS or SMC messages in the SYSLOG or SMC job log.

SMC0090 `Unable to restart subtask [SMCOCMD|SMCFDRV|SMCBMID|SMCJSWP]`

Message Level: 0

Explanation: During subsystem processing the indicated SMC subsystem task abended and could not be restarted.

System Action: Processing continues. However, subsystem processing or command facilities may be affected.

User Response: Stop and restart the SMC subsystem. If the indicated subtask is SMCOCMD, use the MVS CANCEL command to terminate the subsystem.

SMC0091 `Could not allocate job JJJJJJJJ step SSSSSSSS after applying all exclusion levels; backing up until allocatable`

Message Level: 8

Explanation: The indicated job step would not allocate when all exclusion criteria were applied to all DDs in the step. This means that the remaining set of drives is insufficient for each DD to be assigned a unique drive.

System Action: The SMC backs out exclusion criteria on selected DDs in the step until the set of remaining drives is sufficient to allocate to all DDs in the step.

User Response: None.

SMC0095 `UX01 function code UX01REPLY not supported`

Message Level: 4

Explanation: The function code return of UX01RPLY (reply to a WTOR message) is not supported by SMC.

System Action: None. The user exit is ignored.

User Response: None.

SMC0096 `Invalid UX01 function code X'xx'`

Message Level: 4

Explanation: A user exit 01 returned an invalid function code of X'*xx*'.

System Action: All data returned by this invocation of UX01 is ignored.

User Response: Correct the UX01 to return only function codes documented in the *MVS/CSC* or *HSC System Programmer's Guide*.

SMC0097 `UX01 function code X'xx' not valid for message MMMMMMMM`

Message Level: 4

Explanation: The SMC detected that the action code returned for a message (mount, dismount, or swap) did not match the action for an SMC default message.

System Action: All data returned by this invocation of UX01 is ignored.

User Response: Correct the UX01 to return a function code compatible with the message.

SMC0098 `UX01 drive XXXX does not match message drive XXXX, using UX01 drive`

Message Level: 4

Explanation: The device address returned from UX01 does not match the address specified in the message.

System Action: The device address returned from UX01 is used in the message processing.

User Response: Ensure that UX01 is functioning as desired.

SMC0099 `Drive XXXX is not a library drive`

Message Level: 4

Explanation: A user exit 01 returned a drive *XXXX*, which is not a library or virtual drive.

System Action: All data returned by this invocation of UX01 is ignored.

User Response: Ensure that UX01 is functioning as desired.

SMC0100 `Invalid UX01 drive XXXX`

Message Level: 4

Explanation: A user exit 01 returned an invalid drive address of *XXXX* which is not defined as a valid tape device on this system.

System Action: All data returned by this invocation of UX01 is ignored.

User Response: Correct the UX01.

SMC0101 `Invalid UX01 or TAPEREQ subpool` *`SSSSSSSSSSSS`*

Message Level: 4

Explanation: A user exit 01 or TAPEREQ specified an invalid subpool name.

System Action: The default subpool 0 is used.

User Response: Correct the UX01 or TAPEREQ to specify a valid subpool.

SMC0102 `Invalid UX01 volser` *`volser`*

Message Level: 4

Explanation: A user exit 01 returned a volume serial containing invalid characters.

System Action: The volume serial returned by this invocation of UX01 is ignored.

User Response: Correct the UX01 to return a valid volume serial.

SMC0103 `Unrecoverable mount error on device` *`XXXX`* `volser` *`volser`* `for JOB` *`JJJJJJJJ`*

Message Level: 0

Explanation: The SMC detected a volume mount error in response to an IAT5310 message for a mount requested in message IAT5210.

System Action: The SMC breaks the mount loop and issues a dismount to the indicated device *XXXX*. The job remains in the MDS VERIFY queue.

User Response: Refer to the IAT5310 message for the cause of the mount error and take corrective action.

SMC0104 `Default recording technique set for device` *`XXXX`*

Message Level: 0

Explanation: During initialization of the SMC drive table, a device with an unrecognized recording technique was detected.

System Action: A recording technique is defaulted based on the UCB device type. Processing continues.

User Response: Verify that any maintenance for new device types has been applied to all NCS products, including SMC.

SMC0106 `Mount of HSC volser volser on non-HSC device XXXX`

Message Level: 0

Explanation: An HSC subsystem on the host specified a VOLWATCH parameter on the MNTD command. The SMC has detected that volume resident in the HSC library is being requested to be mounted on a drive outside the HSC library.

System Action: None.

User Response: Cancel the job requesting the mount, or eject the volume to satisfy the mount request.

SMC0107 `Swap volser from XXXX1 to XXXX2`

Message Level: 4

Explanation: The SMC has intercepted DDR swap processing. *XXXX2* is either the original device selected to swap to or a device selected by SMC that more closely matches the device characteristics of *XXXX1*.

System Action: DDR swap processing continues.

User Response: None.

SMC0108 `No compatible drive found for SWAP processing`

Message Level: 0

Explanation: The SMC has intercepted DDR swap processing. The original device chosen to swap to is not compatible with the swap from device and the SMC could not locate a compatible alternate device.

System Action: DDR swap processing continues.

User Response: If no compatible device is available, reply NO to message IGF500D or IGF509D to stop the swap process. If a compatible device is offline, vary it online and reply with its device number to message IGF500D or IGF509D.

SMC0109 `The SMC subsystem is running in key n; results are unpredictable; reply 'Y' to continue or 'N' to terminate`

Message Level: 0

Explanation: During initialization SMC detected that it is running in key *n*, not 1 through 7.

System Action: The SMC waits until a reply is received.

User Response: A reply of N stops SMC. A reply of Y causes SMC to continue initialization, though results are unpredictable. Possible problems are S0C1 and S0C4 ABENDs. To prevent this message, update the Program Properties Table (PPT) with "PPT PGMNAME(SMCBINT),SYST,KEY(*n*)", where *n* is between 1 and 7, inclusive.

SMC0110 `Allow swap of volser from XXXX1 to XXXX2; Reply 'Y', 'N' or DEVICE`

Message Level: 0

Explanation: The SMC has intercepted DDR swap processing. The SMC is awaiting operator approval to allow the swap.

System Action: The SMC continues processing; however, the swap cannot complete until an operator reply is entered.

User Response: To allow the swap to proceed using the selected device *XXXX2*, reply Y. To select a different swap to device, reply with its device address. The SMC does not validate a new device address. To cancel the swap, reply N. If an I/O error on device *XXXX1* caused the swap, a reply of N causes the job to fail.

SMC0111 `USER defined message ID: MMMMMMMM`

Message Level: 0

Explanation: The SMC has listed the messages that were added using the USERMsg command.

System Action: Both SMC default and USERMsg added messages are sent to user exit 01 for each library subsystem. SMC processes USERMsg added messages according to the response from user exit 01.

User Response: None.

SMC0112 `Cannot add duplicate message ID MMMMMMMM at line nnnn of SMCPARMS`

Message Level: 0

Explanation: The USERMsg command was entered to add a new message ID that SMC will intercept. The supplied message ID *MMMMMMMM* is a duplicate of a message already defined to SMC.

System Action: The message is not added.

User Response: None.

SMC9999 `MMMMMMMM Variable text`

Message Level: 12, 16, 20, 24

Explanation: SMC9999 messages are intended for StorageTek Technical Support problem determination and resolution. *MMMMMMMM* is the name of the issuing module.

System Action: None.

User Response: None. A message level (LVL) of 12 or higher should generally be specified only when directed by StorageTek Technical Support.

SMCU0001 `SMC is not active or not JES3; unable to produce report`

Explanation: The SMCUPJS utility was submitted on a system without an active SMC subsystem, or on a system that is not JES3.

System Action: Report processing terminates with a return code of 8.

User Response: If the host system is JES3, start the SMC subsystem and resubmit the utility job.

SMCU0002 `Utility release level n.n is incompatible with SMC release n.n`

Explanation: The SMCUPJS utility load module is not at the same release level as the SMC subsystem on the host.

System Action: Report processing terminates with a return code of 8.

User Response: Resubmit the utility job with the release level matching the SMC subsystem.

SMCU0003 `No HSC or MVS/CSC active; unable to determine drive characteristics`

Explanation: The SMCUPJS utility was submitted on a system without an active HSC or MVS/CSC subsystem.

System Action: Report processing terminates with a return code of 8.

User Response: Start the HSC or MVS/CSC subsystem, and issue the SMC RESYNC command. Then resubmit the utility job.

SMCU0004 `*** WARNING: HCD esoteric not found`

Explanation: The SMCUPJS utility found an esoteric defined by the JES3 SETUNIT statement that had no corresponding HCD esoteric.

System Action: Report processing continues, but a return code of 4 is returned.

User Response: Research and correct the discrepancy.

SMCU0005 `*** WARNING: HCD esoteric does not match JES3 esoteric`

Explanation: The SMCUPJS utility found an esoteric that contains different devices in its HCD and JES3 definitions.

System Action: The report lists JES3 devices missing from the HCD esoteric and HCD devices missing from the JES3 esoteric. Report processing continues, but a return code of 4 is returned.

User Response: Research and correct the discrepancy.

SMCU0006 `*** WARNING: XTYPE contains inconsistent location or drive characteristics`

Explanation: Drives contained within an XTYPE do not have the same location type (library, virtual, nonlibrary, or unknown), the same location (ACS or VTSS), or the same recording technique.

System Action: Report processing continues, but a return code of 4 is returned.

User Response: Review the Device to XTYPE report to determine the inconsistency, and correct the discrepancy.

SMCU0007 `*** WARNING:  XTYPE contains unknown or MODEL(IGNORE) devices`

Explanation: One or more of the drives within an XTYPE are either UNKNOWN, MODEL(IGNORE), or both. However, one or more drives in the XTYPE are NOT UNKNOWN or MODEL(IGNORE).

System Action: Report processing continues, but a return code of 4 is returned.

User Response: Verify that the XTYPEs are defined as intended.

Appendix A. JES2 User Exit Return Codes and Library Subsystem Ownership

Specific Volume User Exit 08

For specific volume requests, the HSC User Exit SLSUX08 and MVS/CSC User Exit SCSUX08 can determine ownership. Table 1 and Table 2 describe subsystem request ownership based on User Exit 08 return codes.

Table 1. Subsystem Ownership – HSC User Exit 08

SLSUX08 Return Code	Ownership Decision
0 (Honor JCL)	No decision.
4 (Esoteric returned)	HSC is the owner if the esoteric returned contains at least one HSC-owned drive. No decision if the esoteric returned does not contain any drives owned by the HSC.
8 (Select SPECVOL ACSid(s))	HSC is the owner.
12 (Select nonlibrary drives)	No decision.
16 (ACSid returned)	HSC is the owner.
64 (Inactive)	No decision.

Table 2. Subsystem Ownership - MVS/CSC User Exit 08

SCSUX08 Return Code	Ownership Decision
0 (Honor JCL)	No decision.
4 (Select Library Drives)	This MVS/CSC is the owner.
8 (Select Nonlib Drives)	No decision.
12 (Esoteric returned)	This MVS/CSC is the owner if the esoteric returned contains at least one drive owned by this MVS/CSC. No decision if the esoteric returned does not contain any drives owned by this MVS/CSC.
16 (ACSid and/or CSC name returned)	This MVS/CSC is the owner if the ACSid returns without a different MVS/CSC name. The MVS/CSC named in UX08CSCN is the owner unless it is not active.
64 (Inactive)	No decision.

Scratch Volume User Exit 02

For scratch volume requests, HSC User Exit SLSUX02 and MVS/CSC User Exit SCSUX02 can determine ownership. Table 3 and Table 4 describe subsystem request ownership based on User Exit 02 return codes.

Table 3. Subsystem Ownership – HSC User Exit 02

SLSUX02 Return Code	Ownership Decision
0 (Default allocation)	No decision.
4 (Select library drives)	HSC is the owner.
8 (Select nonlibrary drives)	No decision.
12 (Prefer library drives)	HSC is the owner.
16 (Esoteric returned)	HSC is the owner if the esoteric returned contains at least one HSC owned drive. No decision if the esoteric returned does not contain any drives owned by the HSC.
20 (ACSid returned)	HSC is the owner.
24 (Change LSMPRE value)	HSC is the owner.
32 (Select virtual media)	HSC is the owner.
64 (Inactive)	No decision.

Table 4. Subsystem Ownership – MVS/CSC User Exit 02

SCSUX02 Return Code	Ownership Decision
0 (Normal CSC)	No decision.
4 (Select Library Drives)	This MVS/CSC is the owner.
8 (Select Nonlib Drives)	No decision.
12 (Prefer Library Drives)	This MVS/CSC is the owner.
16 (Esoteric returned)	This MVS/CSC is the owner if the esoteric returned contains at least drive owned by this MVS/CSC. No decision if the esoteric returned does not contain any drives owned by this MVS/CSC.
20 (ACSid and/or CSC name returned)	This MVS/CSC is the owner if the ACSid returns without a different MVS/CSC name. The MVS/CSC named in UX02CSCN is the owner unless it is not active.
24 (Use virtual media)	This MVS/CSC is the owner.
64 (Inactive)	No decision.

For example, a scratch allocation request can be directed to library drives owned by the MVS/CSC even if the HSC has eligible scratch volumes in its library:

```
//DD1 DD UNIT=349Ø,DSN=NEW.DATASET,DISP=(NEW,KEEP)
```

Assume the following:

- HSC and MVS/CSC are both active.
- HSC SLSUX02 returns 0.
- MVS/CSC SCSUX02 returns 4.

The SLSUX02 return code did not decide ownership. The MVS/CSC user exit is called, and its return code indicates its ownership of the allocation.

Appendix B. JES3 User Exit Return Codes and Library Subsystem Ownership

Specific Volume User Exit 13

For specific volume requests, HSC User Exit SLSUX13 and MVS/CSC User Exit SCSUX13 can determine ownership. Table 1 and Table 2 describe subsystem request ownership based on User Exit 13 return codes.

Table 1. Subsystem Ownership – HSC User Exit 13

SLSUX13 Return Code	Ownership Decision
0 (Normal allocation)	No decision.
4 (Select library drives)	HSC is the owner.
8 (Select nonlibrary drives)	No decision.
12 (Esoteric returned)	HSC is the owner if the esoteric returned contains at least one HSC-owned drive. No decision if the esoteric returned does not contain any drives owned by the HSC.
16 (ACSid returned)	HSC is the owner.
64 (Inactive)	No decision.

Table 2. Subsystem Ownership – MVS/CSC User Exit 13

SCSUX13 Return Code	Ownership Decision
0 (Honor JCL)	No decision.
4 (Select Library Drives)	This MVS/CSC is the owner.
8 (Select Nonlib Drives)	No decision.
12 (Esoteric returned)	This MVS/CSC is the owner if the esoteric returned contains at least one drive owned by this MVS/CSC. No decision if the esoteric returned does not contain any drives owned by this MVS/CSC.
16 (ACSid and/or CSC name returned)	This MVS/CSC is the owner if the ACSid returns without a different MVS/CSC name. The MVS/CSC named in UX13CSCN is the owner unless it is not active.
64 (Inactive)	No decision.

Scratch Volume User Exit 04

For scratch volume requests, HSC User Exit SLSUX04 and MVS/CSC User Exit SCSUX04 can determine ownership. Table 3 and Table 4 describe subsystem request ownership based on User Exit 04 return codes.

Table 3. Subsystem Ownership – HSC User Exit 04

SLSUX04 Return Code	Ownership Decision
0 (Default allocation)	No decision.
4 (Select library drives)	HSC is the owner.
8 (Select nonlibrary drives)	No decision.
12 (Esoteric returned)	HSC is the owner if the esoteric returned contains at least one HSC owned drive. No decision if the esoteric returned does not contain any drives owned by the HSC.
16 (ACSid returned)	HSC is the owner.
20 (Change LSMPRE value)	HSC is the owner.
24 (Select virtual media)	HSC is the owner.
64 (Inactive)	No decision.

Table 4. Subsystem Ownership – MVS/CSC User Exit 04

SCSUX04 Return Code	Ownership Decision
0 (Normal CSC)	No decision.
4 (Select Library Drives)	This MVS/CSC is the owner.
8 (Select Nonlib Drives)	No decision.
12 (Esoteric returned)	This MVS/CSC is the owner if the esoteric returned contains at least drive owned by this MVS/CSC. No decision if the esoteric returned does not contain any drives owned by this MVS/CSC.
16 (ACSid and/or CSC name returned)	This MVS/CSC is the owner if the ACSid returns without a different MVS/CSC name. The MVS/CSC named in UX13CSCN is the owner unless it is not active.
24 (Use virtual media)	This MVS/CSC is the owner.
64 (Inactive)	No decision.

For example, a scratch allocation request can be directed to library drives owned by the MVS/CSC even if the HSC has eligible scratch volumes in its library:

```
//DD1 DD UNIT=349Ø,DSN=NEW.DATASET,DISP=(NEW,KEEP)
```

Assume the following:

- HSC and MVS/CSC are both active.
- HSC SLSUX04 returns 0.
- MVS/CSC SCSUX04 returns 4.

The SLSUX04 return code did not decide ownership. The MVS/CSC user exit is called, and its return code indicates its ownership of the allocation.

Appendix C. Intercepted Messages

IBM Operating System Messages

The messages listed in Table 5 are received by the SMC. IBM message manuals describe the exact format (spacing, etc.) and definition for each message. Ellipses are used to indicate that the message contains more text than is shown.

Correct operation of the SMC depends on these messages. Do not suppress or alter them using products designed to handle messages through the subsystem interface (SSI). Many automated operations systems make use of the subsystem interface to intercept and alter or suppress messages.

Note: If the ‘‘suppressed by subsystem’’ and ‘‘hardcopy only’’ bits are turned on in the WQE (the MVS write queue element) before the SMC has received the message, the SMC ignores the WTO, and the message is not displayed at the console.

If you plan to use an automated operations system and are unsure about how it intercepts messages, contact the product vendor.

Although these messages may be suppressed (that is, prevented from displaying in the console) by using MPFLSTxx parameters or an MPF exit, the text of these messages should not be changed. The use of other WTO exits to change the display characteristics or the text of these messages is not supported by the SMC.

Volume serial numbers (‘ser’) as specified in messages from the operating system, are defined as follows:

SCRTCH:	a public scratch volume is to be mounted (nonspecific request).
PRIVAT:	a private scratch volume is to be mounted (nonspecific request).
xxxxxx:	volume serial number to be processed (specific request)

Messages that contain VOLSERs with more than six characters or any character except A-Z, 0-9, # (crosshatch), $, ¥ (yen character), and optional trailing blanks are ignored by the SMC.

Table 5. Intercepted Operating System Messages

Message ID	Description
IEC068A	U dddd,ser
IEC101A	M dddd,ser,...
IEC111E	D dddd,ser
IEC114E	D dddd...
IEC135A	U dddd,ser...
IEC400A	M dddd, ser...
IEC401A	F dddd,ser...
IEC501A	M dddd,ser{,labtyp}
IEC501E	M dddd,ser{,labtyp}
IEC502E	n,dddd,ser...
IEC509A	F dddd,ser...
IEC512I	I/O ERR|LBL ERR|SEC VOL...
IEC701D	M dddd, VOLUME TO BE LABELED ser
IEC702I	dddd, VOLUME LABELS CANNOT BE VERIFIED
IEC703I	dddd, VOLUME IS FILE PROTECTED
IEF233A	M dddd,ser{,labtyp}
IEF233D	M dddd,ser{,labtyp}
IEF234E	{K|D|R} dddd{,ser...}
IGF500I	SWAP dddd to eeee - OPERATOR|I/O ERROR
IGF502E	PROCEED WITH SWAP OF dddd TO eeee
IGF503I	ERROR ON dddd, SELECT NEW DEVICE
IGF509I	SWAP ddd - OPERATOR|I/O ERROR
IGF511A	WRONG VOLUME MOUNTED ON dddd, MOUNT ser,...
_TA0233D	Message for ASM2

JES3 Messages

The following JES3 messages are processed by the SMC:

- IAT5210
- IAT5310
- IAT5410

IBM message manuals describe the exact format (spacing, etc.) and definition of each message.

Tape Management System Messages

CA-1 Messages

The following CA-1 (TMS) messages are received by the SMC. Refer to the appropriate Computer Associates publication for the exact format and meaning of each message.

Table 6. Tape Management System Messages - CA-1

Message ID	Description
CTS001	See CA-1 User Manual, Volume 1
CTS002	See CA-1 User Manual, Volume 1
CTS004	See CA-1 User Manual, Volume 1
CTS005	See CA-1 User Manual, Volume 1
CTS007	See CA-1 User Manual, Volume 1
CTS008	See CA-1 User Manual, Volume 1
CTS009	See CA-1 User Manual, Volume 1
CTS010	See CA-1 User Manual, Volume 1
CTS011	See CA-1 User Manual, Volume 1
CTS014	See CA-1 User Manual, Volume 1
CTS015	See CA-1 User Manual, Volume 1
CTT100A	See CA-1 User Manual, Volume 1
CTT101A	See CA-1 User Manual, Volume 1
CTT102A	See CA-1 User Manual, Volume 1
CTT103A	See CA-1 User Manual, Volume 1
CTT104A	See CA-1 User Manual, Volume 1
CTT105A	See CA-1 User Manual, Volume 1
TMS001	See CA-1 User Manual, Volume 1
TMS002	See CA-1 User Manual, Volume 1

Table 6. Tape Management System Messages - CA-1 (Continued)

Message ID	Description
TMS004	See CA-1 User Manual, Volume 1
TMS005	See CA-1 User Manual, Volume 1
TMS007	See CA-1 User Manual, Volume 1
TMS008	See CA-1 User Manual, Volume 1
TMS009	See CA-1 User Manual, Volume 1
TMS010	See CA-1 User Manual, Volume 1
TMS011	See CA-1 User Manual, Volume 1
TMS014	See CA-1 User Manual, Volume 1
TMS015	See CA-1 User Manual, Volume 1
IECTMS3	See CA-1 User Manual, Volume 1
IECTMS7	See CA-1 User Manual, Volume 1
CA$F810A	See CA-1 Message Guide
CA$F813A	See CA-1 Message Guide

CONTROL-M/TAPE (formerly CONTROL-T) Messages

The following CONTROL-M/TAPE messages are received by the SMC. Refer to the appropriate BMC publication for the exact format and meaning of each message.

Message ID	Description
CTT100A	See BMC's INCONTROL for OS/390 and z/OS Message Manual
CTT101A	See BMC's INCONTROL for OS/390 and z/OS Message Manual
CTT102A	See BMC's INCONTROL for OS/390 and z/OS Message Manual
CTT103A	See BMC's INCONTROL for OS/390 and z/OS Message Manual
CTT104A	See BMC's INCONTROL for OS/390 and z/OS Message Manual
CTT105A	See BMC's INCONTROL for OS/390 and z/OS Message Manual

DFSMSrmm Messages

DFSMSrmm mount message (EDG6627A) must be acted on by the SMC when the volume or drive specified in the message is under the control of the SMC. The action of the SMC is similar to the SMC actions for the normal MVS mount messages (e.g., IEC233A, etc).

The DFSMSrmm Tape Initialization program (EDGINERS) issues a series of messages describing the success or failure of tape initialization, erasure, and/or verification. This series of messages is used to drive the dismount of the tapes mounted from the EDG6627A message. Refer to Table 7 for the messages that must be acted on by the SMC to dismount a tape.

Table 7. Tape Management System Messages - DFSMSrmm

Message ID	Description
EDG6620I	VOLUME volser INITIALIZATION AND VERIFICATION SUCCESSFUL
EDG6621E	VOLUME volser INITIALIZATION FAILED
EDG6623I	VOLUME volser ERASE, INITIALIZATION AND VERIFICATION SUCCESSFUL
EDG6624I	VOLUME volser ERASE FAILED
EDG6627E	M dev VOLUME (volser) RACK (rack-number) TO BE action, lbltype
EDG6642E	VOLUME volser LABELLED SUCCESSFULLY
EDG6643E	VOLUME volser ERASED AND LABELLED SUCCESSFULLY

Appendix D. Message Change Summary

The following messages have been added, changed, or deleted for SMC Release 5.1.

New Messages

- SMC0051
- SMC0060
- SMC0091
- SMC0094
- SMC0095
- SMC0096
- SMC0097
- SMC0098
- SMC0099
- SMC0100
- SMC0101
- SMC0102
- SMC0103
- SMC0104
- SMC0106
- SMC0107
- SMC0108
- SMC0109
- SMC0110
- SMC0111
- SMC0112
- SMCU0001
- SMCU0002
- SMCU0003
- SMCU0004
- SMCU0005
- SMCU0006
- SMCU0007

Changed Messages

- SMC0001
- SMC0003
- SMC0012
- SMC0013
- SMC0025
- SMC0030
- SMC0066
- SMC0077
- SMC0089
- SMC0090

Deleted Messages

- SMC0067
- SMC0094

Index

A

C

D

E

F

H

I

J

L

M

O

P

R

S

T

U

V

X

Z

Printed in U.S.A.